Final Report for AOARD Grant 104053

"**Learning to Predict Social Influence in Complex Networks**"

29/03/2012

Name of Principal Investigators: Kazumi Saito

- e-mail address : k-saito@u-shizuoka-ken.ac.jp
- Institution : School of Administration and Informatics, University of Shizuoka
- Mailing Address : 52-1 Yada, Suruga-ku, Shizuoka 422-8526 Japan
- Phone : +81-54-264-5436
- Fax : +81-54-264-5436

Period of Performance: 18/03/2010 – 17/03/2012

Abstract:

First, we addressed the problem of analyzing information diffusion process in a social network using two kinds of information diffusion models, incorporating asynchronous time delay, the AsIC (Asynchronous time Independent Cascade) model and the AsLT (Asynchronous time Linear Threshold) model, and investigated how the results differ according to the model used. To this end, we developed novel methods of selecting models that better explains the observation. In our behavioral analysis of topic propagation using the real blog propagation data, we found that there is a clear indication as to which topic better follows which model.

Second, we addressed the problem of how different opinions with different values spread over a social network and how their share changes over time in a machine learning setting using a variant of voter model, the value-weighted voter model with multiple opinions. The task is first to estimate the opinion values from the limited amount of observed data and the goal is to predict the expected opinion share at a future target time. We derived an algorithm that guarantees the global optimal solution for the opinion value estimation and the share prediction results outperformed a simple linear extrapolation approximation when the available data is limited.

Third, we addressed the problem of detecting the change in opinion share over a social network caused by an unknown external situation change under the value-weighted voter (VwV) model with multiple opinions in a retrospective setting. We solved this problem by iteratively maximizing the likelihood of generating the observed opinion share, and in doing so we devised a very efficient search algorithm which avoids parameter value optimization during the search. We confirmed that the algorithm can efficiently identify the change and outperforms the naive method, in which an exhaustive search is deployed, both in terms of accuracy and computation time.

Fourth, we addressed the problem of estimating changes in diffusion probability over a social network from the observed information diffusion results, by focusing on the AsIC model in the SIS (Susceptible/Infected/Susceptible) setting. We assumed that the diffusion parameter changes are approximated by a series of step functions, and their changes are reflected in the observed diffusion results. Thus, the problem is reduced to detecting how many step functions are needed, where in time each one starts and how long it lasts, and what the height of each one is. The method employs the derivative of the likelihood function of the observed data that are assumed to be generated from the AsIC-SIS model, adopts a divide-and-conquer type greedy recursive partitioning, and utilizes an MDL model selection measure to determine the adequate number of step functions. The results obtained using real world network structures confirmed that the method works well as intended.

Fifth, we proposed an opinion formation model, an extension of the voter model that incorporates the strength of each node, which is modeled as a function of the node attributes. Then, we addressed the problem of estimating parameter values for these attributes that appear in the function from the observed opinion formation data and solve this by maximizing the likelihood using an iterative parameter value updating algorithm, which is efficient and is guaranteed to converge. We showed that the proposed algorithm can correctly learn the dependency in our experiments on four real world networks for which we used the assumed attribute dependency. We further showed that the influence degree of each node based on the extended voter model is substantially different from that obtained assuming a uniform strength, and is more sensitive to the node strength than the node degree even for a moderate value of the node strength.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
11 APR 2012	**Final**	**18-03-2010 to 17-03-2012**

4. TITLE AND SUBTITLE
Learning to Predict Social Influence in Complex Networks

5a. CONTRACT NUMBER
FA2386-10-1-4053

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Saito Kazumi

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
School of Administration and Informatics,University of Shizuoka,52-1 Yada, Suruga-ku,Shizuoka 422-8526 Japan,NA,NA

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AOARD, UNIT 45002, APO, AP, 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)
AOARD

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AOARD-104053

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this project the following five major achievement are made. 1) two kinds of information diffusion models incorporating asynchronous time dekay and a method to select models that better explains the observation, 2) a method to learn and predict opinion share using a variant of voter model, 3) a method to detect changes in opinion share, 4) a method to detect changes in diffusion probability, and 5) a method to learn the strength of opinion. Each of them uses probabilistic models and machine learning techniques to learn the model parameters from the observation. These are important steps to construct basic methods for learning to predict social influence in complex networks. All of them have been published in international conferences and/or international journals. In total there are 17 publications and they are included in the final report.

15. SUBJECT TERMS
Information diffusion, Opinion formation, Social network, Outbreak detection, Influential nodes, Network dynamics, Knowledge discovery from network

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**289**	

Introduction:

Social entities like human constantly receive/give social influence through interactions, and it evolves over time. Such social interaction processes are usually characterized by highly distributed phenomena in complex networks, but the complexity and distributed nature of those processes does not imply that these evolutions are chaotic or unpredictable. Just as natural scientists discover laws and create models for their fields, so can one, in principle, find empirical regularities and develop explanatory accounts of changes in complex networks. Especially, such predictive knowledge would be valuable for anticipating social trends, and market opportunities. The ultimate goal of our research project is to develop computational methods for learning to predict future activities or behaviors of social entities as social influence in complex networks.

Experiment:

In our experiments, we employed four datasets of large real networks (all bidirectionally connected), based on which information/opinion diffusion data were generated synthetically. The first one is a trackback network of Japanese blogs, and has 12,047 nodes and 79,920 directed links. The second one is a network of people derived from the "list of people" within Japanese Wikipedia, and has 9,481 nodes and 245,044 directed links. The third one is a network derived from the Enron Email Dataset by extracting the senders and the recipients and linking those that had bidirectional communications. It has 4,254 nodes and 44,314 directed links. The fourth one is a coauthorship network, and has 12,357 nodes and 38,896 directed links. In these experiments, anonymized data were utilized and we focused mainly on predicting statistical macroscopic information such as opinion shares as an instance of social influence. Significant parts of these experiments were done by personal computers with main memory of 64 Giga Bite, which were purchased by using the AOARD grant for our previous project.

Results and Discussion:

Model selection: We experimentally confirmed that the learning algorithms converge to the correct values very stably for both the AsLT model and the AsIC mdoel, and the model selection method can correctly identify the diffusion models by which the observed data sets are generated based on extensive simulations on four real world datasets. We further applied the methods to the real blog data and analyzed the behavior of topic propagation. The relative propagation speed of topics, i.e. how far/near and how fast/slow each topic propagates, that are derived from the learned parameter values is rather insensitive to the model selected, but the model selection algorithm clearly identifies the difference of model goodness for each topic. We found that many of the topics follow the AsIC models in general, but some specific topics have clear interpretations for them being better modeled by either one of the two, and these interpretations are consistent with the model selection results. There are numerous factors that affect the information diffusion process, and there can be a number of different models. Model selection is a big challenge in social network analysis and this work is the first step towards this goal.

Behavioral analysis: We experimentally confirmed that the learning algorithm converges to a correct solution and predicts the expected opinion share over a social network at a target time from the opinion diffusion data under the value-weighted voter model with multiple opinions. We also showed using two real world social networks that the values are learnable from a small amount of observed data and the share prediction with use of the estimated values is satisfactorily accurate and outperforms the prediction by a simple linear extrapolation. Theoretical analysis for a situation where the local opinion share can be approximated by the average opinion share over the whole network, (e.g., the case of a complete network), revealed that the expected share prediction problem is well-defined only when the opinion values are non-uniform in which case the final consensus is winners-takes-all, i.e., the opinion with the highest value wins and all the others die, and when they are uniform, any opinion can be a winner. Our immediate future work is to validate the credibility of the voter model using available real opinion propagation data.

Change-point detection for VwV model: We assumed that the opinion value changed by unknown external factors and focused on the form of change of a rect-linear pattern, that is, the value changes to a new higher level, persists for a certain period of time (hot span), and is restored back to the original level and stays the same thereafter. In order to detect such a change pattern, the naive learning algorithm has to iteratively update the pattern boundaries (outer loop) and the opinion value must also be optimized for each combination of the pattern boundaries (inner loop), which is extraordinary inefficient. For this problem, we devised a very efficient search algorithm which avoids

the inner loop optimization during the search. We tested the performance using the structures of four real world networks, and confirmed that the algorithm can efficiently identify the hot span correctly as well as the opinion value. We further compared our algorithm with the naive method that finds the best combination of change boundaries by an exhaustive search through a set of randomly selected boundary candidates, and showed that the proposed algorithm far outperforms the native method both in terms of accuracy and computation time.

Change-points detection for AsIC-SIS model: We tested our algorithm to artificially generated change patterns using four real world network structures. The results obtained confirmed that the method works well as intended. The algorithm is efficient because it needs to do expensive parameter optimization only once for each partitioning (which is not that many in many cases). The MDL criterion is useful to avoid overfitting. In many cases it identifies the correct number of step functions, but in some cases the found pattern is not necessarily the same in terms of the number of step functions, but the error is always reduced to a small value. The found pattern is not necessarily the same in terms of the number of step functions as the one assumed to be true, but the error is always reduced to a small value.

Super-influential node extraction: We tested our model and algorithm on four real world networks assuming the attribute dependency of the parameters to be of a particular form. We showed that the algorithm can correctly estimate the strength of each node by way of node attributes through a learned function. We further showed that the influence degree of each node based on our model is substantially different from a naive model that assumes a uniform strength throughout the nodes for which the influence degree is known to be proportional to the node degree, and there appears to be no simple heuristic to approximate the influence degree with good accuracy unless the network is dense. The sensitivity analysis indicated that as the degree of non-uniformity of the strength becomes greater, the influence degree becomes progressively more sensitive to the node strength than the node degree, and even for a moderate value of the non-uniformity of node strength, it is more affected by the node strength than by the node degree.

During the research period, as scheduled in our original proposal, we analyzed and developed several models for information diffusion and opinion formation, which served as basic models for exploring the successive research tasks. Namely, we have taken some important steps along the path to construct basic methods for learning to predict social influence in complex networks as a major result of Year 1. In Year 2, we developed very promising core methods for change-points detection in model parameters. We also developed the other methods applicable to such tasks including future outbreaks prediction from early observation and super-influential nodes extraction in social networks, and evaluated each method by using synthetic diffusion data sets. We believe that our proposed models and methods will play an important role to discover new insights into social influence in a wide variety of societies.

List of Publications: List any publications, conference presentations, or patents that resulted from this work.

1. Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda, "Efficient Discovery of Influential Nodes for SIS Models in Social Networks," Knowledge and Information Systems: An International Journal (Online First:10.1007/s10115-011-0396-2), 2011.
2. Akihiro Koide, Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda, "Estimating Diffusion Probability Changes for AsIC-SIS Model from Information Diffusion Results," The third Asian Conference on Machine Learning (ACML2011), pp.297--313, 2011
3. Yuki Yamagishi, Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda, "Learning Attribute-weighted Voter Model over Social Networks," The third Asian Conference on Machine Learning (ACML2011), pp.263--280, 2011.
4. Masahiro Kimura, Kazumi Saito, Kouzou Ohara, and Hiroshi Motoda, "Detecting Anti-majority Opinionists Using Value-weighted Mixture Voter Model," Proc. of the 14th International Conference on Discovery Science (DS2011) , pp. 150–164, 2011.
5. Kouzou Ohara, Kazumi Saito, Masahiro Kimura, and Hiroshi Motoda, "Efficient Detection of Hot Span in Information Diffusion from Observation," Proc. of the IJCAI-2011 Workshop on Link Analysis in Heterogeneous Information Networks (IJCAI-HINA2011), arXiv:1110.2659v1, 2011.
6. Kazumi Saito, Kouzou Ohara, Yuki Yamagishi, Masahiro Kimura, and Hiroshi Motoda, "Learning Diffusion Probability based on Node Attributes in Social Networks," Proc. of the 19th

International Symposium on Methodologies for Intelligent Systems (ISMIS2011), pp. 153–162, 2011.
7. Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda, "Detecting Changes in Opinion Value Distribution for Voter Model," Proc. of the 2011 International Conference on Social Computing, Behavioral Modeling, and Prediction (SBP2011), pp. 89--96, 2011.
8. Masahiro Kimura, Kazumi Saito, Kouzou Ohara, and Hiroshi Motoda, "Learning Information Diffusion Model in a Social Network for Predicting Influence of Nodes," Intelligent Data Analysis - An International Journal, Vol.15, No.4, pp.633--652, 2011.
9. Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda, "Generative Models of Information Diffusion with Asynchronous Time-delay," The second Asian Conference on Machine Learning (ACML2010), pp.203--218, 2010.
10. Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda, "Discovery of Super-Mediators of Information Diffusion in Social Networks," Proc. of the 13th International Conference on Discovery Science (DS2010), LANAI6332, pp.144--158, 2010.
11. Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda, "Selecting Information Diffusion Models over Social Networks for Behavioral Analysis," Proc. of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2010), pp.180-195, 2010.
12. Yuya Yoshikawa, Kazumi Saito, Hiroshi Motoda, Kouzou Ohara, and Masahiro Kimura "Acquiring Expected Influence Curve from Single Diffusion Sequence," Proc. of the 2010 Pacific Rim Knowledge Acquisition Workshop (PKAW2010), pp.275-289, 2010.
13. Takayasu Fushimi, Kazumi Saito, Masahiro Kimura, Hiroshi Motoda, and Kouzou Ohara "Finding Relation between PageRank and Voter Model," Proc. of the 2010 Pacific Rim Knowledge Acquisition Workshop (PKAW2010), pp.209-223, 2010.
14. Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda, "Efficient Estimation of Cumulative Influence for Multiple Activation Information Diffusion Model with Continuous Time Delay," Proc. of the Eleventh Pacific Rim International Conference on Artificial Intelligence (PRICAI2010), pp.244-255, 2010.
15. Masahiro Kimura, Kazumi Saito, Kouzou Ohara, and Hiroshi Motoda, "Learning to Predict Opinion Share in Social Networks," Proc. of the Twenty-Fourth Conference on Artificial Intelligence (AAAI2010), pp.1364--1370, 2010.
16. Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda, "Behavioral Analyses of Information Diffusion Models by Observed Data of Social Network," Proc. of the 2010 International Conference on Social Computing, Behavioral Modeling, and Prediction (SBP2010), pp.149-158, 2010.
17. Masahiro Kimura, Kazumi Saito, Ryohei Nakano, and Hiroshi Motoda, "Extracting Influential Nodes on a Social Network for Information Diffusion," Data Mining and Knowledge Discovery, Vol.20, No.1, pp.70--97, 2010.

Attachments: Publications listed above.

Under consideration for publication in Knowledge and Information Systems

Efficient Discovery of Influential Nodes for SIS Models in Social Networks

Kazumi Saito[1], Masahiro Kimura[2], Kouzou Ohara[3], and Hiroshi Motoda[4]

[1]School of Administration and Informatics, University of Shizuoka, Shizuoka 422-8526, Japan;
[2]Department of Electronics and Informatics, Ryukoku University, Otsu 520-2194, Japan;
[3]Department of Integrated Information Technology, Aoyama Gakuin University, Kanagawa 229-8558, Japan;
[4]Institute of Scientific and Industrial Research, Osaka University, Osaka 567-0047, Japan

Abstract. We address the problem of discovering the influential nodes in a social network under the *susceptible/infected/susceptible (SIS) model* which allows multiple activation of the same node, by defining two influence maximization problems:*final-time* and *integral-time*. We solve this problem by constructing a layered graph from the original network with each layer added on top as the time proceeds and applying the bond percolation with two effective control strategies: pruning and burnout. We experimentally demonstrate that the proposed method gives much better solutions than the conventional methods that are based solely on the notion of centrality using two real-world networks. The pruning is most effective when searching for a single influential node, but burnout is more powerful in searching for multiple nodes which together are influential. We further show that the computational complexity is much smaller than the naive probabilistic simulation both by theory and experiment. The influential nodes discovered are substantially different from those identified by the centrality measures. We further note that the solutions of the two optimization problems are also substantially different, indicating the importance of distinguishing these two problem characteristics and using the right objective function that best suits the task in hand.

Keywords: Information diffusion; SIS model; Influence maximization; Pruning method; Burnout method

1. Introduction

Social networks mediate the spread of various information including topics, ideas and even (computer) viruses. The proliferation of emails, blogs and social networking services (SNS) in the World Wide Web accelerates the creation of large social networks.

Received May 22, 2010
Revised Mar 01, 2011
Accepted Mar 20, 2011

Therefore, substantial attention has recently been directed to investigating information diffusion phenomena in social networks (Newman, 2001; Adar and Adamic, 2005; Domingos, 2005; McCallum et al, 2005; Leskovec et al, 2007b; Watts and Dodds, 2007; Agarwal and Liu, 2008), and other aspects such as analyses of social networking sites (Mislove et al, 2007; Muhlestein and Lim, 2009), topic evolution (Zhou et al, 2006; Peng and Li, 2010), and privacy issues (Backstrom et al, 2007; Zhou and Pei, 2010).

Finding influential nodes is one of the central problems in social network analysis[1]. Thus, developing efficient and practical methods of doing this on the basis of information diffusion is an important research issue. Widely used fundamental probabilistic models of information diffusion are the *independent cascade (IC) model* (Goldenberg et al, 2001; Kempe et al, 2003; Gruhl et al, 2004) and the *linear threshold (LT) model* (Watts, 2002; Kempe et al, 2003). Researchers investigated the problem of finding a limited number of influential nodes that are effective for the spread of information under the above models (Kempe et al, 2003; Kimura et al, 2007; Kimura et al, 2010). This combinatorial optimization problem is called the *influence maximization problem*. Kempe et al (2003) experimentally showed on large collaboration networks that the greedy algorithm can give a good approximate solution to this problem, and mathematically proved a performance guarantee of the greedy solution (i.e., the solution obtained by the greedy algorithm). Recently, methods based on bond percolation (Kimura et al, 2007) and submodularity (Leskovec et al, 2007a) were proposed for efficiently estimating the greedy solution. Succeeding work further improved the efficiency by approximating the solution using a heuristic (Chen et al, 2009). The influence maximization problem has applications in sociology and "viral marketing" (Agarwal and Liu, 2008), and was also investigated in a different setting (a descriptive probabilistic model of interaction) (Domingos and Richardson, 2001; Richardson and Domingos, 2002). The problem has recently been extended to influence control problems such as a contamination minimization problem (Kimura et al, 2009a).

The IC model can be identified with the so-called *susceptible/infected/recovered (SIR) model* for the spread of a disease (Newman, 2003; Gruhl et al, 2004). In the SIR model, only infected individuals can infect susceptible individuals, while recovered individuals can neither infect others nor be infected by others. This implies that an individual is never infected with the disease multiple times. This property holds true for the LT model as well. However, there are many phenomena for which this property does not hold. A typical example would be the following propagation phenomenon of a topic in the blogosphere: A blogger who has not yet posted a message about the topic is interested in the topic by reading the blog of a friend, and posts a message about it (i.e., becoming infected (activated)[2]). Next, the same blogger reads a new message about the topic posted by some other friend, and may post a message (i.e., becoming infected) again. Note here that we regard the act of "posting" to be the state change from "susceptible" to "infected". The blogger can read the next blog and respond to it anytime after the completion of the previous posting. Most simply, this phenomenon can be modeled by a *susceptible/infected/susceptible (SIS) model* from the epidemiology. Other examples include the growth of hyper-link posts among bloggers (Leskovec et al, 2007b), the spread of computer viruses without permanent virus-checking programs, and epidemic disease such as tuberculosis and gonorrhea (Newman, 2003). There are

[1] "Influence" means many things and there are many factors which make a node influential. In this paper, as we describe later in this section and define more formally in subsection 2.2, influence of a node simply means the expected number of activated nodes as a result of information diffusion that starts from the node.

[2] We use "infected" and "activated" interchangeably.

many more examples of information diffusion phenomena for which the SIS model is more appropriate.

We focus on an information diffusion process in a social network $G = (V, E)$ over a given time span T on the basis of an SIS model. Here, the SIS model is a stochastic process model, and the *influence* of a set of nodes H at time-step t, $\sigma(H, t)$, is defined as the expected number of infected nodes at time-step t when all the nodes in H are initially infected at time-step $t = 0$. We refer to σ as the *influence function* for the SIS model. When we want to find an influential node, we need to know $\sigma(\{v\}, t)$, ($v \in V$, $t = 1, \ldots, T$), but when we want to solve influence maximization problem, we need to know $\sigma(H, t)$, ($H \subseteq V$, $t = 1, \ldots, T$). It is vital, first of all, to have an effective method for estimating $\sigma(\{v\}, t)$. Clearly, in order to extract influential nodes, we must estimate the value of $\sigma(\{v\}, t)$ for every node v and every time-step t. Solving influence maximization problem is much more difficult because we have to find the optimal subset of nods H^*_K with a fixed cardinality K. Here it is vital to have an effective method for evaluating the *marginal influence gains* $\{\sigma(H \cup \{v\}, T) - \sigma(H, T); v \in V \setminus H\}$ for any non-empty subset H of V. We have reported our preliminary work on efficiently estimating $\{\sigma(\{v\}, t); v \in V, t = 1, \ldots, T\}$ for the SIS model based on the bond percolation with a pruning strategy (Kimura et al, 2009b), and extended it to influential maximization problem in which we introduced a new technique called burnout to efficiently estimate $\{\sigma(H \cup \{v\}, T) - \sigma(H, T); v \in V \setminus H\}$ (Saito et al, 2009).

In this paper, we describe these two techniques in details and conduct extensive experiments to evaluate how these two affect the efficiency of solving the influence maximization problems on a network $G = (V, E)$ under the SIS model. Needless to say, we can naively estimate the marginal influence gains for any non-empty subset H of V by simulating the SIS model. However, this naive simulation method is overly inefficient and not practical at all. Here, we define two influence maximization problems: the *final-time maximization problem* and the *integral-time maximization problem*. The latter problem does not make sense for the SIR model and is only meaningful for the SIS model. We adopt the greedy algorithm, to reduce the computational complexity, for approximately solving the problems according to the work of Kempe et al (2003) which was conducted for the IC and the LT models, ensuring that submodularity holds in the SIS model setting, too. We show theoretically that the proposed method is expected to achieve a large reduction in computational cost by comparing computational complexity with the naive probabilistic simulation method. Further, using two large real networks, we experimentally demonstrate that the proposed method is much more efficient than the naive greedy method that uses only the bond percolation without employing both the pruning and the burnout. We show that the pruning is effective when searching for a single influential node, but the burnout is more powerful and eventually takes over the pruning as we increase the number of nodes to search. Thus, it is advisable to use both the pruning and the burnout only in the initial few iterations and stop using the pruning and use the burnout alone in the succeeding iterations in the greedy algorithm. The computational cost reduces by 2 orders of magnitudes comparing the naive bond percolation which itself is 2 to 3 orders of magnitudes more efficient than the naive simulation. We also show that the nodes discovered by the proposed method are substantially different from the nodes discovered by the conventional methods that are based on the notion of various centrality measures which does not consider the information diffusion phenomena and can be evaluated from the network topology alone. The proposed method results in a substantial increase in the expected influence. We further find that the two optimization problems give also substantially different solutions and it is important to use the right objective function which reflects the problem characterization.

The paper is organized as follows. We define the information diffusion model in sec-

tion 2 and the two influential maximization problems we want to solve in section 3. We then give details of the algorithms to solve this problem (greedy algorithm, bond percolation, pruning, burnout and their combinations) in section 4. The experimental results are given in section 5 (network data, quality of the solutions and computation time for both influence function estimation and influence maximization estimation), followed by some discussions in section 6. We end this paper by summarizing the conclusion in section 7.

2. Information Diffusion Model

Let $G = (V, E)$ be a directed network, where V and E stand for the sets of all the nodes and (directed) links, respectively. Here, note that E is a subset of $V \times V$. For any $v \in V$, let $\Gamma(v; G)$ denote the set of the child nodes (directed neighbors) of v, that is,

$$\Gamma(v; G) = \{w \in V;\ (v, w) \in E\}.$$

2.1. SIS Model

An SIS model for the spread of a disease is based on the cycle of disease in a host. A person is first *susceptible* to the disease, and becomes *infected* with some probability when the person has contact with an infected person. The infected person becomes susceptible to the disease soon without moving to the immune state. We consider a discrete-time SIS model for information diffusion on a network. In this context, infected nodes mean that they have just adopted the information, and we call these infected nodes *active* nodes.

We define the SIS model for information diffusion on G. In the model, the diffusion process unfolds in discrete time-steps $t \geq 0$, and it is assumed that the state of a node is either active or inactive. For every link $(u, v) \in E$, we specify a real value $p_{u,v}$ with $0 < p_{u,v} < 1$ in advance. Here, $p_{u,v}$ is referred to as the *diffusion probability* through link (u, v). Given an initial set of active nodes H and a time span T, the diffusion process proceeds in the following way. Suppose that node u becomes active at time-step t ($< T$). Then, node u attempts to activate every $v \in \Gamma(u; G)$, and succeeds with probability $p_{u,v}$. If node u succeeds, then node v will become active at time-step $t + 1$. If multiple active nodes attempt to activate node v at time-step t, then their activation attempts are sequenced in an arbitrary order. On the other hand, node u becomes or remains inactive at time-step $t + 1$ unless it is activated from other active node at time-step t. The process terminates if the current time-step reaches the time limit T.

2.2. Influence Function

For the SIS model on G, we consider an information diffusion from an initially activated node set $H \subset V$ over time span T. Let $S(H, t)$ denote the set of active nodes at time-step t. Note that $S(H, t)$ is a random subset of V and $S(H, 0) = H$. Let $\sigma(H, t)$ denote the expected number of $|S(H, t)|$, where $|X|$ stands for the number of elements in a set X. We call $\sigma(H, t)$ the *influence* of node set H at time-step t. Note that σ is a function defined on $2^V \times \{0, 1, \cdots, T\}$. We call the function σ the *influence function* for the SIS model over time span T on network G. In view of more complex social influence, we need to incorporate a number of social factors with social networks such as rank, prestige and power. In our approach, we assume that we can encode such factors as diffusion

probabilities of each node[3]. As emphasized in section 1, it is important to estimate the influence function σ efficiently. In theory we can simply estimate σ by the simulations based on the SIS model in the following way. First, a sufficiently large positive integer M is specified. For each $H \subset V$, the diffusion process of the SIS model is simulated from the initially activated node set H, and the number of active nodes at time-step t, $|S(H,t)|$, is calculated for every $t \in \{0,1,\cdots,T\}$. Then, $\sigma(H,t)$ is estimated as the empirical mean of $|S(H,t)|$'s that are obtained from M such simulations. However, this is extremely inefficient, and cannot be practical.

3. Influence Maximization Problem

We mathematically define the influence maximization problems on a network $G = (V,E)$ under the SIS model. Let K be a positive integer with $K < |V|$. First, we define the *final-time maximization problem*: Find a set H_K^* of K nodes to target for initial activation such that $\sigma(H_K^*;T) \geq \sigma(H;T)$ for any set H of k nodes, that is, find

$$H_K^* = \arg \max_{\{H \subset V;\, |H|=K\}} \sigma(H;T). \tag{1}$$

Second, we define the *integral-time maximization problem*: Find a set H_K^* of K nodes to target for initial activation such that $\sigma(H_K^*;1)+\cdots+\sigma(H_K^*;T) \geq \sigma(H;1)+\cdots+\sigma(H;T)$ for any set H of k nodes, that is, find

$$H_K^* = \arg \max_{\{H \subset V;\, |H|=K\}} \sum_{t=1}^{T} \sigma(H;t). \tag{2}$$

The first problem cares only how many nodes are influenced at the time of interest. For example, in an election campaign it is only those people who are convinced to vote the candidate at the time of voting that really matter and not those who were convinced during the campaign but changed their mind at the very end. Maximizing the number of people who actually vote falls in this category. The second problem cares how many nodes have been influenced throughout the period of interest. For example, maximizing the amount of product purchase during a sales campaign falls in this category.

4. Proposed Method

Kempe et al (2003) showed the effectiveness of the greedy algorithm for the influence maximization problem under the IC and LT models. In this section, we introduce the greedy algorithm for the SIS model, and describe three techniques (the bond percolation method, the pruning method, and the burnout method) for efficiently solving the influence maximization problem under the greedy algorithm. We also discuss the computational complexity of these methods and show the merit of the pruning and the burnout.

[3] Such factors as rank, prestige and power exert influence in a cumulative way, i.e. richer gets richer phenomena. We need some reinforcement mechanism outside the SIS model to deal with such feedback which is beyond the scope of our framework.

4.1. Greedy Algorithm

We approximately solve the influence maximization problem by the greedy algorithm. Below we describe this algorithm first for the final-time maximization problem and then for the integral-time maximization problem.

Greedy algorithm for the final-time maximization problem:

$\mathcal{A}$**1.** Set $H \leftarrow \emptyset$.

$\mathcal{A}$**2.** For $k = 1$ to K do the following steps:

$\mathcal{A}$**2-1.** Choose a node $v_k \in V \setminus H$ maximizing $\sigma(H \cup \{v\}, T)$.

$\mathcal{A}$**2-2.** Set $H \leftarrow H \cup \{v_k\}$.

$\mathcal{A}$**3.** Output H.

We can easily modify this algorithm for the integral-time maximization problem by replacing step $\mathcal{A}$2-1 as follows:

Greedy algorithm for the integral-time maximization problem:

$\mathcal{A}$**1.** Set $H \leftarrow \emptyset$.

$\mathcal{A}$**2.** For $k = 1$ to K do the following steps:

$\mathcal{A}$**2-1'.** Choose a node $v_k \in V \setminus H$ maximizing $\sum_{t=1}^{T} \sigma(H \cup \{v\}, t)$.

$\mathcal{A}$**2-2.** Set $H \leftarrow H \cup \{v_k\}$.

$\mathcal{A}$**3.** Output H.

Let H_K denote the set of K nodes obtained by this algorithm. We refer to H_K as the *greedy solution* of size K. Then, it is known that

$$\sigma(H_K, t) \;\geq\; \left(1 - \frac{1}{e}\right) \sigma(H_K^*, t),$$

where H_k^* is the exact solution defined by Equation (1) or (2), that is, the expected influence of the greedy solution is lower bounded and it is guaranteed that it is at worst 63% of the optimal expected influence (Kempe et al, 2003).

To implement the greedy algorithm, we need a method for estimating all the marginal influence degrees $\{\sigma(H \cup \{v\}, t);\, v \in V \setminus H\}$ of H in step $\mathcal{A}$2-1 or $\mathcal{A}$2-1' of the above algorithms. In the subsequent subsections, we propose a method for efficiently estimating the influence function σ over time span T for the SIS model on network G.

4.2. Layered Graph

We build a layered graph $G^T = (V^T, E^T)$ from G in the following way (see Figure 1). First, for each node $v \in V$ and each time-step $t \in \{0, 1, \cdots, T\}$, we generate a copy v_t of v at time-step t. Let V_t denote the set of copies of all $v \in V$ at time-step t. We define V^T by $V^T = V_0 \cup V_1 \cup \cdots \cup V_T$. In particular, we identify V with V_0. Next, for each link $(u, v) \in E$, we generate T links (u_{t-1}, v_t), $(t \in \{1, \cdots, T\})$, in the set of nodes V^T. We set $E_t = \{(u_{t-1}, v_t);\, (u, v) \in E\}$, and define E^T by $E^T = E_1 \cup \cdots \cup E_T$. Moreover, for any link (u_{t-1}, v_t) of the layered graph G^T, we define the occupation probability q_{u_{t-1}, v_t} by $q_{u_{t-1}, v_t} = p_{u,v}$.

Then, we can easily prove that the SIS model with diffusion probabilities $\{p_e; e \in E\}$ on G over time span T is equivalent to the *bond percolation process (BP) with occu-*

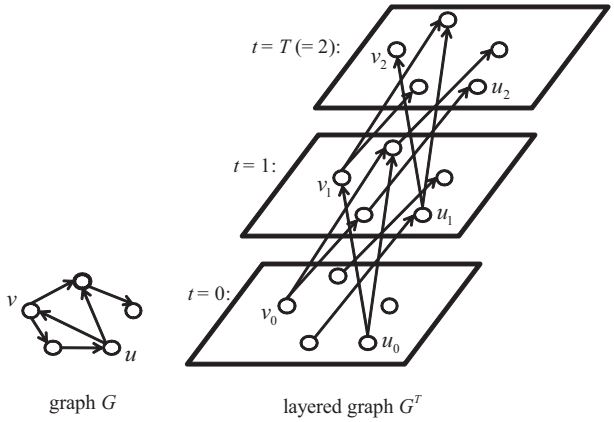


Fig. 1. An example of a layered graph.

pation probabilities $\{q_e; e \in E^T\}$ on G^T.[4] Here, the BP process with occupation probabilities $\{q_e; e \in E^T\}$ on G^T is the random process in which each link $e \in E^T$ is independently declared "occupied" with probability q_e. We perform the BP process on G^T, and generate a graph constructed by occupied links, $\tilde{G}^T = (V^T, \tilde{E}^T)$. Then, in terms of information diffusion by the SIS model on G, an occupied link $(u_{t-1}, v_t) \in E_t$ represents a link $(u, v) \in E$ through which the information propagates at time-step t, and an unoccupied link $(u_{t-1}, v_t) \in E_t$ represents a link $(u, v) \in E$ through which the information does not propagate at time-step t. For any $v \in V \setminus H$, let $F(H \cup \{v\}; \tilde{G}^T)$ be the set of all nodes that can be reached from $H \cup \{v\} \in V_0$ through a path on the graph $\tilde{G}^T$. When we consider a diffusion sample from an initial active node $v \in V$ for the SIS model on G, $F(H \cup \{v\}; \tilde{G}^T) \cap V_t$ represents the set of active nodes at time-step t, $S(H \cup \{v\}, t)$.

4.3. Bond Percolation Method

Using the equivalent BP process, we present a method for efficiently estimating influence function σ. We refer to this method as the *BP method*. Unlike the naive method, the BP method simultaneously estimates $\sigma(H \cup \{v\}, t)$ for all $v \in V \setminus H$. Moreover, the BP method does not fully perform the BP process, but performs it partially. Note first that all the paths from nodes $H \cup \{v\}$ ($v \in V \setminus H$) on the graph $\tilde{G}^T$ represent a diffusion sample from the initial active nodes $H \cup \{v\}$ for the SIS model on G. Let L' be the set of the links in G^T that start from the non-activated nodes in the diffusion sample. For calculating $|S(H \cup \{v\}, t)|$, it is unnecessary to determine whether the links in L' are occupied or not. Therefore, the BP method performs the BP process for only an appropriate set of links in G^T. The BP method estimates σ by the following algorithm:

BP method:

$\mathcal{B}$**1.** Set $\sigma(H \cup \{v\}, t) \leftarrow 0$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$.

$\mathcal{B}$**2.** Repeat the following procedure M times:

[4] The SIS model over time span T on G can be exactly mapped onto the IC model on G^T (Kempe et al, 2003). Thus, the result follows from the equivalence of the BP process and the IC model (Grassberger, 1983; Newman, 2002; Kempe et al, 2003; Kimura et al, 2007).

$\mathcal{B}$**2-1.** Initialize $S(H \cup \{v\}, 0) = H \cup \{v\}$ for each $v \in V \setminus H$, and set $A(0) \leftarrow V \setminus H$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$.

$\mathcal{B}$**2-2.** For $t = 1$ to T do the following steps:

$\mathcal{B}$**2-2a.** Compute $B(t-1) = \bigcup_{v \in A(t-1)} S(H \cup \{v\}, t-1)$.

$\mathcal{B}$**2-2b.** Perform the BP process for the links from $B(t-1)$ in G^T, and generate the graph $\tilde{G}_t$ constructed by the occupied links.

$\mathcal{B}$**2-2c.** For each $v \in A(t-1)$, compute $S(H \cup \{v\}, t) = \bigcup_{w \in S(H \cup \{v\}, t-1)} \Gamma(w; \tilde{G}_t)$, and set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t) + |S(H \cup \{v\}, t)|$ and $A(t) \leftarrow A(t) \cup \{v\}$ if $S(H \cup \{v\}, t) \neq \emptyset$.

$\mathcal{B}$**3.** For each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$, set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t)/M$, and output $\sigma(H \cup \{v\}, t)$.

Note that $A(t)$ finally becomes the set of information source nodes that have at least an active node at time-step t, that is, $A(t) = \{v \in V \setminus H;\ S(H \cup \{v\}, t) \neq \emptyset\}$. Note also that $B(t-1)$ is the set of nodes that are activated at time-step $t-1$ by some source nodes, that is, $B(t-1) = \bigcup_{v \in V} S(H \cup \{v\}, t-1)$.

Now we estimate the computational complexity of the BP method in terms of the number of the nodes, $\mathcal{N}_a$, that are identified in step $\mathcal{B}$2-2a, the number of the coin-flips, $\mathcal{N}_b$, for the BP process in step $\mathcal{B}$2-2b, and the number of the links, $\mathcal{N}_c$, that are followed in step $\mathcal{B}$2-2c. Let $d(v)$ be the number of out-links from node v (i.e., out-degree of v) and $d'(v)$ the average number of occupied out-links from node v after the BP process. Here we can estimate $d'(v)$ by $\sum_{w \in \Gamma(v;G)} p_{v,w}$. Then, for each time-step $t \in \{1, \cdots, T\}$, we have

$$\mathcal{N}_a = \sum_{v \in A(t-1)} |S(H \cup \{v\}, t-1)|, \quad \mathcal{N}_b = \sum_{w \in B(t-1)} d(w), \quad \mathcal{N}_c = \sum_{v \in A(t-1)} \sum_{w \in S(H \cup \{v\}, t-1)} d'(w) \quad (3)$$

on the average.

In order to compare the computational complexity of the BP method to that of the naive method, we consider mapping the naive method onto the BP framework, that is, separating the coin-flip process and the link-following process. We can easily verify that the following algorithm in the BP framework is equivalent to the naive method:

Naive method expressed in the framework of BP method:

$\mathcal{B}$**1.** Set $\sigma(H \cup \{v\}, t) \leftarrow 0$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$.

$\mathcal{B}$**2.** Repeat the following procedure M times:

$\mathcal{B}$**2-1.** Initialize $S(H \cup \{v\}, 0) = H \cup \{v\}$ for each $v \in V \setminus H$, and set $A(0) \leftarrow V \setminus H$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$.

$\mathcal{B}$**2-2.** For $t = 1$ to T do the following steps:

$\mathcal{B}$**2-2b'.** For each $v \in A(t-1)$, perform the BP process for the links from $S(H \cup \{v\}, t-1)$ in G^T, and generate the graph $\tilde{G}_t(v)$ constructed by the occupied links.

$\mathcal{B}$**2-2c'.** For each $v \in A(t-1)$, compute $S(H \cup \{v\}; t) = \bigcup_{w \in S(H \cup \{v\}, t-1)} \Gamma(w; \tilde{G}_t(v))$, and set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t) + |S(H \cup \{v\}, t)|$ and $A(t) \leftarrow A(t) \cup \{v\}$ if $S(H \cup \{v\}, t) \neq \emptyset$.

$\mathcal{B}$**3.** For each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$, set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t)/M$, and output $\sigma(H \cup \{v\}, t)$.

Then, for each $t \in \{1, \cdots, T\}$, the number of coin-flips, $\mathcal{N}_{b'}$, in step $\mathcal{B}$2-2b' is

$$\mathcal{N}_{b'} = \sum_{v \in A(t-1)} \sum_{w \in S(H \cup \{v\}, t-1)} d(w), \quad (4)$$

and the number of the links, $\mathcal{N}_{c'}$, followed in step $\mathcal{B}$2-2c' is equal to $\mathcal{N}_c$ in the BP method on the average. From equations (3) and (4), we can see that $\mathcal{N}_{b'}$ is much larger

than $\mathcal{N}_{c'} = \mathcal{N}_c$, especially for the case where the diffusion probabilities are small. We can also see that $\mathcal{N}_{b'}$ is generally much larger than each of $\mathcal{N}_a$ and $\mathcal{N}_b$ in the BP method for a real social network. In fact, since such a network generally includes large clique-like subgraphs, there are many nodes $w \in V$ such that $d(w) \gg 1$, and we can expect that $\sum_{v \in A(t-1)} |S(H \cup \{v\}, t-1)| \gg |\bigcup_{v \in A(t-1)} S(H \cup \{v\}, t-1)|$ $(= |B(t-1)|)$. Therefore, the BP method is expected to achieve a large reduction in computational cost.

4.4. Pruning Method

In order to further improve the computational efficiency of the BP method, we introduce a pruning technique and propose a method referred to as the *BP with pruning method*. The key idea of the pruning technique is to utilize the following property: Once we have $S(H \cup \{u\}, t_0) = S(H \cup \{v\}, t_0)$ at some time-step t_0 on the course of the BP process for a pair of information source nodes, u and v, then we have $S(H \cup \{u\}, t) = S(H \cup \{v\}, t)$ for all $t > t_0$. The BP with pruning method estimates σ by the following algorithm:

BP with pruning method:

$\mathcal{B}$**1.** Set $\sigma(H \cup \{v\}, t) \leftarrow 0$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$.

$\mathcal{B}$**2.** Repeat the following procedure M times:

$\mathcal{B}$**2-1".** Initialize $S(H \cup \{v\}; 0) = H \cup \{v\}$ for each $v \in V \setminus H$, and set $A(0) \leftarrow V \setminus H$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$, and $C(v) \leftarrow \{v\}$ for each $v \in V \setminus H$.

$\mathcal{B}$**2-2.** For $t = 1$ to T do the following steps:

$\mathcal{B}$**2-2a.** Compute $B(t-1) = \bigcup_{v \in A(t-1)} S(H \cup \{v\}, t-1)$.

$\mathcal{B}$**2-2b.** Perform the BP process for the links from $B(t-1)$ in G^T, and generate the graph $\tilde{G}_t$ constructed by the occupied links.

$\mathcal{B}$**2-2c".** For each $v \in A(t-1)$, compute $S(H \cup \{v\}, t) = \bigcup_{w \in S(H \cup \{v\}, t-1)} \Gamma(w; \tilde{G}_t)$, set $A(t) \leftarrow A(t) \cup \{v\}$ if $S(H \cup \{v\}, t) \neq \emptyset$, and set $\sigma(H \cup \{u\}, t) \leftarrow \sigma(H \cup \{u\}, t) + |S(H \cup \{v\}, t)|$ for each $u \in C(v)$.

$\mathcal{B}$**2-2d.** Check whether $S(H \cup \{u\}, t) = S(H \cup \{v\}, t)$ for $u, v \in A(t)$, and set $C(v) \leftarrow C(v) \cup C(u)$ and $A(t) \leftarrow A(t) \setminus \{u\}$ if $S(H \cup \{u\}, t) = S(H \cup \{v\}, t)$.

$\mathcal{B}$**3.** For each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$, set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t)/M$, and output $\sigma(H \cup \{v\}, t)$.

Basically, by introducing step $\mathcal{B}$2-2d and reducing the size of $A(t)$, the proposed method attempts to improve the computational efficiency over the original BP method. For the proposed method, it is important to implement efficiently the equivalence check process in step $\mathcal{B}$2-2d. In our implementation, we first scan each $v \in A(t)$ according to the value of $n = |S(H \cup \{v\}, t)|$, and identify those nodes with the same n value.

4.5. Burnout Method

In order to further improve the computational efficiency of the BP with pruning method, we introduce another technique called burnout and propose a method which is referred to as the *BP with pruning and burnout method* [5]. More specifically, we focus on the fact that maximizing the marginal influence degree $\sigma(H \cup \{v\}, t)$ with respect to $v \in$

[5] Here we integrated these two techniques, but it is also possible to combine the BP method with only the burnout method. We skipped this one because it is self-evident.

$V \setminus H$ is equivalent to maximizing the marginal influence gain $\phi_H(v,t) = \sigma(H \cup \{v\}, t) - \sigma(H,t)$. Here in terms of the BP process for a newly added information source node v, maximizing $\phi_H(v,t)$ reduces to maximizing $|S(H \cup \{v\}, t) \setminus S(H,t)|$ on the average. The BP with pruning and burnout method estimates ϕ_H by the following algorithm:

BP with pruning and burnout methods:

*C***1.** Set $\phi_H(v,t) \leftarrow 0$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$.

*C***2.** Repeat the following procedure M times:

*C***2-1.** Initialize $S(H;0) = H$, and $S(\{v\};0) = \{v\}$ for each $v \in V \setminus H$, and set $A(0) \leftarrow V \setminus H$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$, and $C(v) \leftarrow \{v\}$ for each $v \in V \setminus H$.

*C***2-2.** For $t = 1$ to T do the following steps:

*C***2-2a.** Compute $B(t-1) = \bigcup_{v \in A(t-1)} S(\{v\}, t-1) \cup S(H, t-1)$.

*C***2-2b.** Perform the BP process for the links from $B(t-1)$ in G^T, and generate the graph $\tilde{G}_t$ constructed by the occupied links.

*C***2-2c.** Compute $S(H,t) = \bigcup_{w \in S(H,t-1)} \Gamma(w; \tilde{G}_t)$, and for each $v \in A(t-1)$, compute $S(\{v\},t) = \bigcup_{w \in S(\{v\},t-1)} \Gamma(w; \tilde{G}_t) \setminus S(H,t)$, set $A(t) \leftarrow A(t) \cup \{v\}$ if $S(\{v\},t) \neq \emptyset$, and set $\phi_H(\{u\},t) \leftarrow \phi_H(\{u\},t) + |S(\{v\},t)|$ for each $u \in C(v)$.

*C***2-2d.** Check whether $S(\{u\},t) = S(\{v\},t)$ for $u, v \in A(t)$, and set $C(v) \leftarrow C(v) \cup C(u)$ and $A(t) \leftarrow A(t) \setminus \{u\}$ if $S(\{u\},t) = S(\{v\},t)$.

*C***3.** For each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$, set $\phi_H(\{v\},t) \leftarrow \phi_H(\{v\},t)/M$, and output $\phi_H(\{v\},t)$.

Intuitively, by using the burnout technique, we can substantially reduce the size of the active node set from $S(H \cup \{v\}, t)$ to $S(\{v\},t)$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$ compared with the BP with pruning method. Namely, in terms of computational costs described by Equation (3), we can expect to obtain smaller numbers for $\mathcal{N}_a$ and $\mathcal{N}_c$ when $H \neq \emptyset$. However, how effectively the proposed method works will depend on several conditions such as network structure, time span, values of diffusion probabilities, etc. We will do a simple analysis later and experimentally show that it is indeed effective.

5. Experimental Evaluation

We have carried out extensive experiments and evaluated the effects of the two techniques that were implemented on top of the bond percolation on the quality of the solution and the computation time, using two real world social networks. The baseline to compare the quality of the solution is the naive simulation method which is confirmed to be prohibitively inefficient.

5.1. Network Data and Basic Settings

In our experiments, we employed two datasets of large real networks used in Kimura et al (2009a), which exhibit many of the key features of social networks (Newman and Park, 2003).

The first one is a trackback network of Japanese blogs. The network data was collected by tracing the trackbacks from one blog in the site "goo (http://blog.goo.ne.jp/)" in May, 2005. We refer to the network data as the blog network. The blog network was a strongly-connected bidirectional network, where a link created by a trackback was regarded as a bidirectional link since blog authors establish mutual communications

by putting trackbacks on each other's blogs. The blog network had $12,047$ nodes and $79,920$ directed links. The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia. Specifically, we extracted the maximal connected component of the undirected graph obtained by linking two people in the "list of people" if they co-occur in six or more Wikipedia pages, and constructed a directed graph by regarding those undirected links as bidirectional ones. We refer to the network data as the Wikipedia network. Thus, the Wikipedia network was also a strongly-connected bidirectional network, and had $9,481$ nodes and $245,044$ directed links.

We assigned a uniform value p to the diffusion probability $p_{u,v}$ for any link $(u,v) \in E$, that is, $p_{u,v} = p$ for the SIS model we used. According to Kempe et al (2003) and Leskovec et al (2007b), we set the value of p relatively small. In particular, we set the value of p to a value smaller than $1/d$, where d is the mean out-degree of a network. Since the values of d were about 6.63 and 25.85 for the blog and the Wikipedia networks, respectively, the corresponding values of $1/d$ were about 0.15 and 0.039. In view of these values we decided to set $p = 0.1$ for the blog network and $p = 0.03$ for the Wikipedia network. Time span T can be arbitrarily set but it is constrained by the inefficiency of the naive simulation method. We found $T = 30$ is good enough to evaluate the performance of our method. We also need to specify the number M of performing the bond percolation process. The larger, the better, but we have to compromise between the solution quality and the computational cost. We set $M = 10,000$ for estimating influence degrees for the blog and Wikipedia networks (See 5.2.1).

All our experimentations were undertaken on a single PC with an Intel Dual Core Xeon X5272 3.4GHz processor, with 32GB of memory, running under Linux.

5.2. Performance for Influence Function Estimation

5.2.1. Accuracy of Estimated Influence Function

We first investigated how accurately the proposed method can estimate the value of influence function in terms of node ranking. Since, in this case, the information diffusion starts with every single node $v \in V$ independently with all the other nodes remaining inactive, i.e. $H = \emptyset$, there is no room for burnout to come in. Thus, we compared the BP with pruning method (BPP for short) with the naive method (naive for short) which we consider as the baseline. Both methods require M to be specified in advance as a parameter. If M is set at ∞, both BPP and naive should give the correct expected influence degree. For a finite value of M, the results may seem different. In fact, as shown in section 4.3, the number of coin flips is different in these two methods and it is much larger in the naive method. However, this does not mean that there is more randomness introduced in the naive method and thus the convergence of the naive method is faster. In fact for each single (initially activated) node v from which to propagate the information, the number of independent coin-flips is effectively the same for both the methods. Thus by using the same value of M, both would estimate $\sigma(v,t)$ with the same accuracy in principle.

We have first experimentally confirmed that use of $M = 100,000$ gives a very stable identical converged solution for both methods for a few selected initial nodes, but the naive method took an order of week to return the result and thus is not practical to perform the comparative study. Then we found that further reducing the value to $M = 10,000$ still gives reliable results, i.e., in effect the same ranking and value of $\sigma(v,t)$, for $t = 1, \cdots, 20$ for the high ranked nodes. The following results were obtained by using

Table 1. Results for the top 10 nodes v and the values of $\sigma(v, 20)$ based on the proposed method (BPP) for the blog network. Left: The result of the first experiment. Right: The result of the second experiment.

Rank	v	$\sigma(v, 20)$	Rank	v	$\sigma(v, 20)$
1	2210	984.74	1	2210	984.87
2	2248	980.41	2	2248	979.46
3	3906	956.97	3	3906	955.84
4	3907	953.04	4	3907	952.71
5	146	929.96	5	146	929.30
6	155	928.77	6	155	928.49
7	3233	912.61	7	3233	911.01
8	3228	912.18	8	3228	910.49
9	140	909.22	9	140	910.31
10	2247	909.12	10	2247	909.59

Table 2. Results for the top 10 nodes v and the values of $\sigma(v, 20)$ based on the naive method for the blog network. Left: The result of the first experiment. Right: The result of the second experiment.

Rank	v	$\sigma(v, 20)$	Rank	v	$\sigma(v, 20)$
1	2210	984.38	1	2210	985.74
2	2248	979.59	2	2248	980.72
3	3906	956.82	3	3906	956.57
4	3907	953.14	4	3907	953.89
5	146	931.03	5	146	931.62
6	155	929.68	6	155	930.21
7	3233	913.50	7	3233	911.89
8	3228	912.27	8	3228	910.52
9	140	910.04	9	140	910.37
10	2247	909.59	10	2247	909.59

$M = 10,000$. Tables 1 and 2 show the ranking of the initially activated influential nodes v evaluated at time-step $T = 20$ for the blog network. We had to limit T to 20 because of the prohibitive computation cost for the naive simulation. The value of influence function $\sigma(v, 20)$ is sorted in the decreasing order and the top 10 nodes are listed. We repeated the experiment twice for each method (BPP and naive) and the results for both are shown side by side. We note that the ranking is exactly the same for the two runs and this is also true between the two methods. We further note that the values of corresponding influence degrees are very similar. The influence degree varies slowly and it decreases only by less than 10% in going from the top to the 10th. Tables 3 and 4 are the results for the Wikipedia network. The results are slightly less stable than for the

Table 3. Results for the top 10 nodes v and the values of $\sigma(v, 20)$ based on the proposed method (BPP) for the Wikipedia network. Left: The result of the first experiment. Right: The result of the second experiment.

Rank	v	$\sigma(v, 20)$	Rank	v	$\sigma(v, 20)$
1	790	2121.52	1	790	2120.45
2	279	2120.52	2	279	2119.32
3	8340	2119.33	3	8340	2118.42
4	323	2118.86	4	323	2117.81
5	326	2117.98	5	326	2117.15
6	772	2117.06	6	772	2116.66
7	325	2116.12	7	325	2114.85
8	2441	2113.09	8	4924	2112.72
9	2465	2112.52	9	1407	2112.44
10	1407	2112.19	10	2498	2111.35

Table 4. Results for the top 10 nodes v and the values of $\sigma(v, 20)$ based on the naive method for the Wikipedia network. Left: The result of the first experiment. Right: The result of the second experiment.

Rank	v	$\sigma(v, 20)$	Rank	v	$\sigma(v, 20)$
1	790	2122.14	1	790	2120.84
2	279	2119.62	2	323	2118.81
3	8340	2119.10	3	279	2118.76
4	323	2117.97	4	8340	2118.52
5	326	2117.84	5	326	2117.75
6	772	2116.37	6	772	2117.32
7	325	2115.84	7	325	2116.39
8	1407	2113.85	8	1407	2114.42
9	4294	2112.79	9	2465	2114.34
10	3149	2112.57	10	4924	2113.55

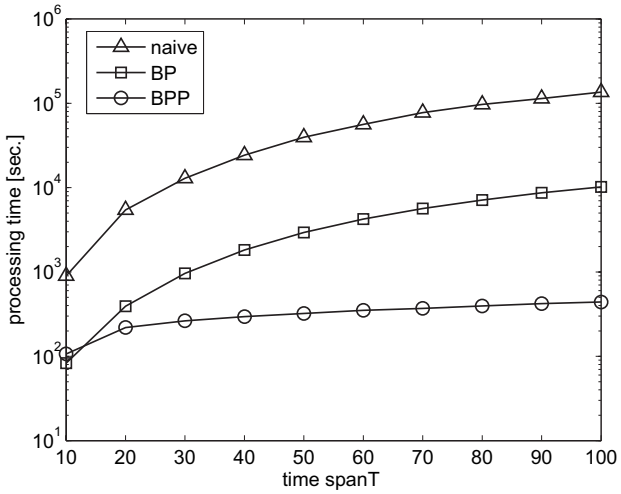


Fig. 2. Results for the blog network.

blog network. However, the rankings of top 7 are the same for the two runs of BPP and the first run of the naive. We note that the values of the influence degrees change much more slowly and the value only reduces by less than 0.5% in going from the top to the 10th. The Wikipedia network is much more difficult in terms of correctly identifying the ranking. From the overall experimental results, we confirm that for the same and large enough values of M, the proposed method (BPP) gives the same results as the naive method.

We have not evaluated the integral influence function over the time span T:$\sum_{t=1}^{T} \sigma(v, t)$ because if it is confirmed that each component $\sigma(v, t)$ can be well approximated, its sum is equally well approximated.

5.2.2. Computational Cost for Influence Function Estimation

Next, we compared the processing time of the proposed method (BPP) with the BP method without pruning (BP for short) and the naive method. Here, we used $M = 1,000$ in order to keep the computational time for the naive method at a reasonable level so that it runs for a larger T. Figures 2 and 3 show the processing time to estimate $\{\sigma(v, t); v \in$

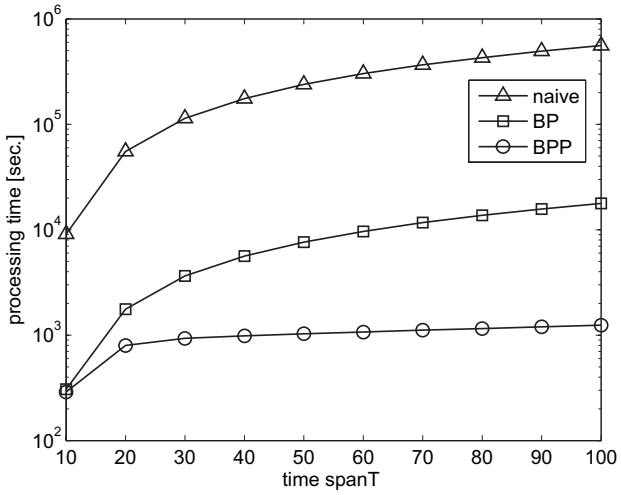


Fig. 3. Results for the Wikipedia network.

$V, t = 0, 1, \cdots, T\}$ as a function of the time span T for the blog and the Wikipedia networks, respectively. In these figures, the circles, squares and triangles indicate the results for BPP, BP and naive, respectively. Note that in case of the blog network, the processing time for the time span $T = 100$ is about 7 minutes, 2.8 hours, and 1.5 days for BPP, BP, and naive, respectively. Namely, BPP is about 25 and 310 times faster than BP and naive, respectively. Note also that in case of the Wikipedia network, the processing time for the time span $T = 100$ is about 21 minutes, 5 hours, and 155 hours for BPP, BP and naive, respectively. Namely, BPP is about 14 and 440 times faster than BP and naive, respectively.

The reduction of the processing time due to the pruning is large. The processing time is about 20 times less when evaluated for $T = 100$. However, when T is small the pruning adversely affects the processing time because of the computational overhead. The two BP methods (with and without pruning) are much faster than the naive method. The performance difference between BPP and each of BP and naive increases as time-step (or time span) increases. Moreover, the same performance difference becomes larger for the blog network than the Wikipedia network. The following simple analysis explains this. Consider the extreme case where $S(u,t) = S(v,t)$ for $\forall u, v \in A(t)$ and $d(w) = d$ for $\forall w \in S(v,t)$ ($v \in A(t)$) at some time-step t. We denote $|A(t)| = a$ and $|S(v,t)| = s$. Then, we have $\mathcal{N}_a = as$, $\mathcal{N}_b = sd$, $\mathcal{N}_{b'} = asd$ and $\mathcal{N}_c = asd'$ on the average for time-step $t+1$. Recall that d' is the expected number of the occupied links, which is calculated as pd, where p is the common diffusion probability for all links. Further assume that the pruning was ideal such that $\tilde{\mathcal{N}}_a = s$ and $\tilde{\mathcal{N}}_c = sd'$, which respectively denote the number of nodes identified in step 2-2a and the average number of links followed in step 2-2c" for BPP. Then, if $ad' > d$, i.e., $ad'/d = ap > 1$ holds, the improvement ratios of BPP over BP and naive are respectively $asd'/sd = ap$ and $asd/sd = a$. From our experimental results, we can estimate a as 310 for the blog network and 440 for the Wikipedia network. Then we obtain ap as 31 and 13 respectively, which approximates the actual ratio ($\text{Proc\_time}_{BP}/\text{Proc\_time}_{BPP}$), 25 and 14. The similar discussion applies to the processing time for the integrated influence function over the time span T.

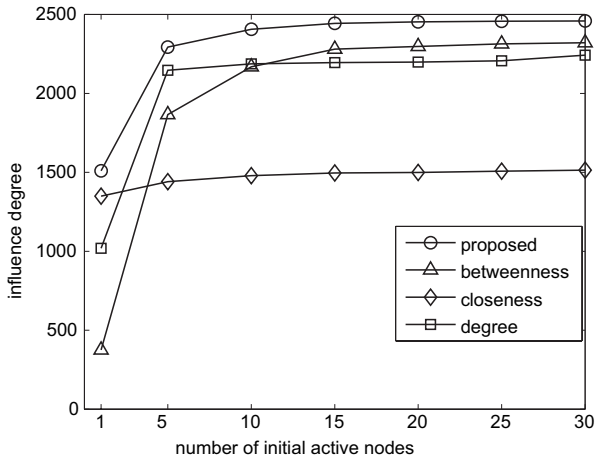


Fig. 4. Comparison of solution quality for the blog network (final-time maximization problem).

5.3. Performance of Influence Maximization Problem

5.3.1. Comparison of Accuracy of the Proposed Methods with Centrality Measures

We compared the quality of the solution of the proposed method, i.e. the BP with pruning and burnout method (BPPB for short) with the three well known centrality measures: "degree centrality", "closeness centrality", and "betweenness centrality" that are commonly used as the influence measure in sociology (Wasserman and Faust, 1994). Here, the betweenness of node *v* is defined as the total number of shortest paths between pairs of nodes that pass through *v*, the closeness of node *v* is defined as the reciprocal of the average distance between *v* and other nodes in the network, and the degree of node *v* is defined as the number of links attached to *v*. We evaluated the value of these measures for each node and ranked the nodes in decreasing order, and calculated the influence degree (both the final-time value and the integral-time value) using the top K nodes with $K = 1, 2, ..., 30$. We refer to these methods as the *betweenness method*, the *closeness method*, and the *degree method*, respectively.

The solution H_K of the proposed method is calculated by the bond percolation algorithm described in 4.5 using both pruning and burnout. Clearly, the quality of H_K can be evaluated by the influence degree $\sigma(H_K, T)$ for the final-time maximization problem and the influence degree $\sum_{t=1}^{T} \sigma(H_K, t)$ for the integral-time maximization problem. We estimated the values of $\sigma(H_K, T)$ and $\sum_{t=1}^{T} \sigma(H_K, t)$ with $M = 10,000$ and $T = 30$. Figures 4 and 5 show the influence degree $\sigma(H_K, T)$ (solution of the final-time maximization problem) as a function of the number of initial active nodes K for the blog and the Wikipedia networks, respectively. In the same way, Figures 6 and 7 show the influence degree $\sum_{t=1}^{T} \sigma(H_K, t)$ (solution of the integral-time maximization problem) as a function of the number of initial active nodes K for the blog and the Wikipedia networks, respectively. In the figures, the circles, triangles, diamonds, and squares indicate the results for the proposed (BPPB), the betweenness, the closeness, and the degree methods, respectively. Evidently, the proposed method performs the best for both networks and for both maximization problems. The shapes of the curves are different for

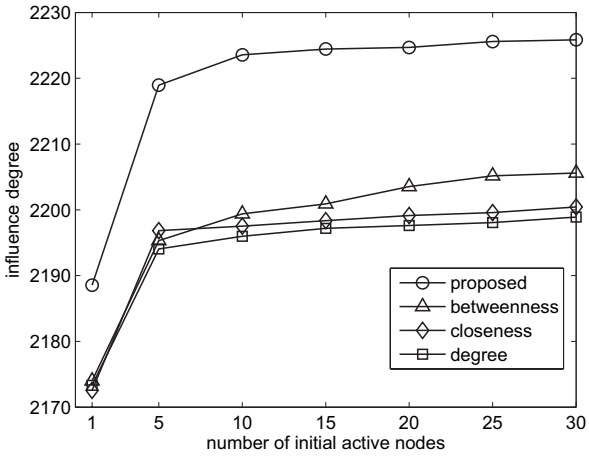


Fig. 5. Comparison of solution quality for the Wikipedia network (final-time maximization problem).

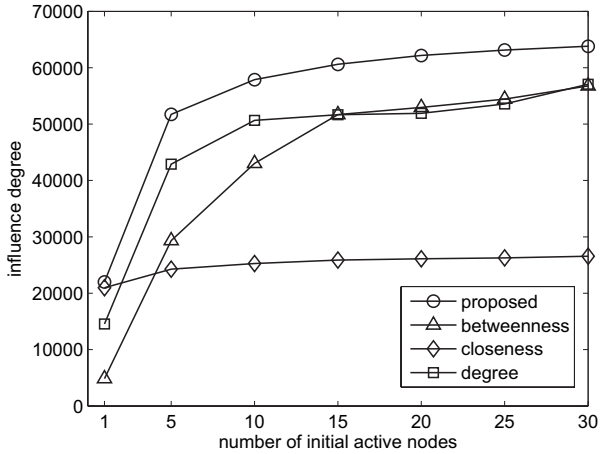


Fig. 6. Comparison of solution quality for the blog network (integral-time maximization problem).

the two problems. In the final-time maximization problem, only the first top 5 to 10 nodes are influential and the succeeding nodes do not contribute to increasing the influence degree. As a rule of thumb, this is true for all the four methods. In the integral-time maximization problem, nodes after the top 10 are also influential and contribute to increasing the influence degree. This is also true for all the four methods as a rule of thumb. There is no clear indication as to which centrality measures rank higher for a wide range of nodes. For example, betweenness measure appears to be the next best for the both networks in case of the final-time maximization problem, but degree measure is also good for the both networks (slightly better for the blog and slightly worse for the Wikipedia network) in case of the integral-time maximization problem. If we focus only the first 10 nodes, degree method appears to be the best among the three conventional

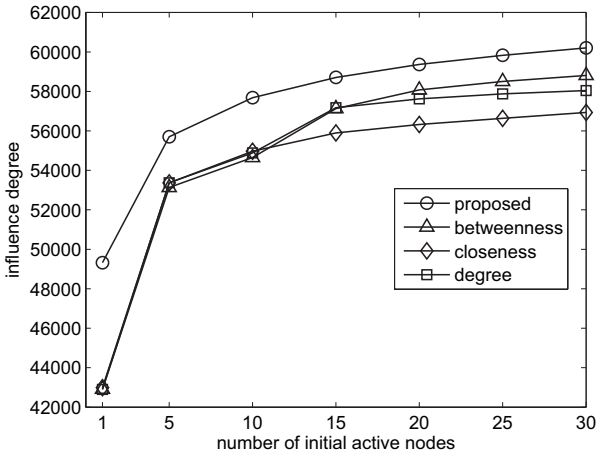


Fig. 7. Comparison of solution quality for the Wikipedia network (integral-time maximization problem).

methods. How well or badly each of the conventional heuristics performs depends on the characteristics of the network structure and the type of the maximization problem. Note that there are substantial differences in the amount of the influence degree (value of the objective function). These results clearly indicate that it is indeed important to obtain the optimal solution. The proposed method can be effectively used for this purpose, and outperforms the conventional heuristics centrality measures from social network analysis.

It is interesting to note that the k nodes ($k = 1, 2, ..., K$) that are discovered to be the most influential by the proposed method are substantially different from those that are found by the conventional centrality measures. For example, in the case of the final-time maximization problem, the best node ($k = 1$) chosen by the proposed method for the blog dataset is ranked 118 for the betweenness method, 659 for the closeness method and 6 for the degree method, and the 15th node ($k = 15$) by the proposed method is ranked 1373, 8848 and 507 for the corresponding conventional methods, respectively. The best node ($k = 1$) chosen by the proposed method for the Wikipedia dataset is ranked 580 for the betweenness method, 2766 for the closeness method and 15 for the degree method, and the 15th node ($k = 15$) by the proposed method is ranked 265, 2041, and 21 for the corresponding conventional methods, respectively. In the case of the integral-time maximization problem, the difference is not that much but is similar by no means. The best node ($k = 1$) chosen by the proposed method for the blog dataset is ranked 17, 5 and 3 for the corresponding conventional methods, and the 15th node ($k = 15$) by the proposed method is ranked 31, 653 and 27, respectively. The best node ($k = 1$) chosen by the proposed method for the Wikipedia dataset is ranked 15, 6 and 3, and the 15th node ($k = 15$) by the proposed method is ranked 84, 23, and 12.

What these results imply is that the influential nodes strongly depend on the objective functions to be maximized, which in turn implies that taking the diffusion process into consideration is crucially important. The results would be affected not only by the network structure but also by the values of diffusion parameters, *i.e.*, even if the network structure remains the same, assigning different diffusion probabilities changes the influence degree of each node. Said differently, any centrality measure that is solely based on network topology has an intrinsic limitation to correctly evaluate the node

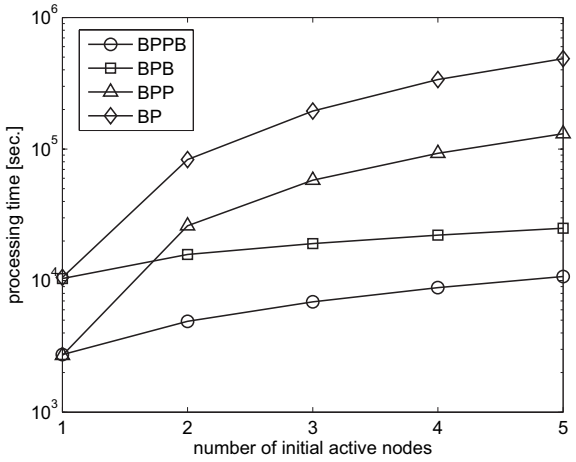


Fig. 8. Comparison of processing time for the blog network (final-time maximization problem).

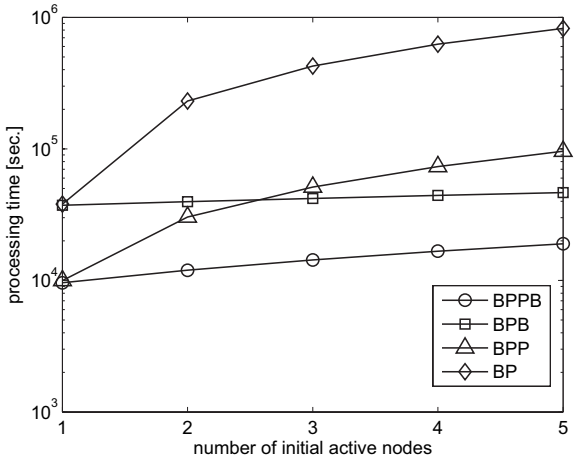


Fig. 9. Comparison of processing time for the Wikipedia network (final-time maximization problem).

influence as defined in this paper. We realize that these centrality measures are not necessarily designed to infer the influential nodes. They have their own advantages, *e.g.*, degree centrality can be used to identify the core nodes of a community and betweenness centrality can be used to study community structure. Indeed, the recently proposed topological centrality (Zhuge and Zhang, 2010) is shown to be very useful to understand the structure of network by distinguishing the roles of nodes, discovering communities and finding underlying backbone networks.

5.3.2. Comparison of Computational Cost among Different Combinations of Component Techniques

Next, we compared the processing time of the proposed method (BPPB) with three other methods with different combinations of component techniques (with/without Pruning and Burnout), i.e. bond percolation only (BP), bond percolation with pruning (BPP) and bond percolation with burnout (BPB) to see the effect of each component. We only show the results for the final-time maximization problem because it is self-evident that the processing time for the integral-time maximization problem is almost the same from the algorithm in 4.1. Figures 8 and 9 show the processing time of these four methods as a function of the number of initial active nodes K for the blog and the Wikipedia networks, respectively. In these figures, circles, triangles, squares and crosses indicate the results of BPPB, BPB, BPP and BP, respectively. The effect of the pruning is shown by the difference of the processing time at $K = 1$ (difference between BP and BPP). The pruning reduces the processing time to about $1/5$, which is consistent with Figs. 2 and 3 for $T = 30$ in 5.2.2. At $K = 2$ the effect of burnout starts appearing and it surpasses the effect of pruning for the blog network (BPB < BPP) but it still does not do so for the Wikipedia network (BPP < BPB). However, after $K \geq 3$ the effect of burnout surpasses the effect of pruning, and burnout plays a key role of reducing the computational cost. Combining the both, i.e., BPPB, always gives the best results within the region where the experiments were performed, i.e. $K \leq 5$. The amount of reduction in processing time by BPPB is large. The processing time of BP and BPPB for $K = 5$ is 5.8 days and 2.8 hours, respectively, for the blog network, and 9.3 days and 5.6 hours, respectively, for the Wikipedia network. The processing time reduces to $1/50$ for the blog network and $1/40$ for the Wikipedia network for $K = 5$. However, it is seen that the difference between BPB and BPPB becomes smaller as K becomes larger and it is predicted that eventually BPB will surpass BPPB, meaning that the overhead of pruning exceeds the saving by pruning. Thus, it is advisable to use both the strategies only in the initial few iterations, and stop using the pruning and use the burnout alone in the succeeding iterations in the greedy algorithm. Note that the above reduction is for $T = 30$. It is expected that the reduction is much larger for a larger T, e.g., $T = 100$, and also for a larger K, e.g. $K = 30$. Needless to say, the naive method needs an order of month to return the results and is prohibitively inefficient. From these results, we can conclude that the proposed method is much more efficient than the simple BP method and can be practical.

6. Discussion

The influence function $\sigma(\cdot, T)$ is submodular (Kempe et al, 2003). For solving a combinatorial optimization problem of a submodular function f on V by the greedy algorithm, Leskovec et al. (Leskovec et al, 2007a) have recently presented a lazy evaluation method that leads to far fewer (expensive) evaluations of the marginal increments $f(H \cup \{v\}) - f(H)$, $(v \in V \setminus H)$ in the greedy algorithm for $H \neq \emptyset$, and achieved an improvement in speed. Note here that their method requires evaluating $f(v)$ for all $v \in V$ at least. Thus, we can apply their method to the influence maximization problem for the SIS model, where the influence function $\sigma(\cdot, T)$ is evaluated by simulating the corresponding random process. It is clear that 1) this method is more efficient than the naive greedy method that does not employ the BP method and instead evaluates the influence degrees by simulating the diffusion phenomena, and 2) further both the methods become the same for $K = 1$ and empirically estimate the influence function

$\sigma(\cdot, T)$ by probabilistic simulations. These methods also require M to be specified in advance as a parameter, where M is the number of simulations. Note that the BP and the simulation methods can estimate influence degree $\sigma(v, t)$ with the same accuracy by using the same value of M. Moreover, estimating influence function $\sigma(\cdot, 30)$ by $10,000$ simulations needed more than 35.8 hours for the blog dataset and 13.2 days for the Wikipedia dataset, respectively. However, the proposed method for $K = 30$ needed less than 7.0 hours for the blog dataset and 13.1 hours for the Wikipedia dataset, respectively. Therefore, it is clear that the proposed method can be faster than the method by Leskovec et al (2007a) for the influence maximization problem for the SIS model. In fact, we have confirmed in Kimura et al (2010) that the bond percolation method is 10 times faster than the lazy evaluation for the SIR model for $K = 30$. Since the SIS model can be mapped to the SIR model by introducing the layered graph, the result above is consistent to our previous result.

We discussed the accuracy and the computational cost of the proposed method in 5.2 and 5.3. Here we look into the solutions of the final-time maximization problem and the integral-time maximization problem. We found that these two different maximization problems give almost totally different nodes although the objective function to be maximized for the latter is the sum of the objective function of the former over the final time T. There is only one common node out of 30 influential nodes in case of the blog network and there are only five common nodes in case of the Wikipedia network. In general the identified influential nodes for the final-time maximization problem reflects the diffusion characteristics of one time slot but those for the integral-time maximization problem reflects the global diffusion characteristics. Intermediate process does not matter and what matters is only the final situation for the former, whereas the whole process does matter for the latter. It is important to distinguish these two different problem characteristics and use the right objective function that best suits the task in hand.

7. Conclusion

Finding influential nodes is one of the most central problems in the field of social network analysis. There are several models that simulate how various things, e.g., news, rumors, diseases, innovation, ideas, etc. diffuse across the network. One such realistic model is the *susceptible/infected/susceptible (SIS) model*, an information diffusion model where nodes are allowed to be activated multiple times. The computational complexity drastically increases because of this multiple activation property, e.g., compared with the *susceptible/infected/recovered (SIR) model* where nodes once activated can never be deactivated/reactivated. We addressed the problem of efficiently discovering the influential nodes under the SIS model, i.e., estimating the expected number of activated nodes at time-step t for $t = 1, \cdots, T$ starting from an initially activated node set $H \in V$ at time-step $t = 0$ and finding the optimal subset H^* to maximize the expected influence. We solved this problem by constructing a layered graph from the original social network by adding each layer on top of the existing layers as the time proceeds, and applying the bond percolation with two control strategies: pruning and burnout. We showed that the computational complexity of the proposed method is much smaller than the conventional naive probabilistic simulation method by a theoretical analysis. We applied the proposed method to two different types of influence maximization problem, i.e. discovering the K most influential nodes that together maximize the expected influence degree at the time of interest (final-time maximization problem) or the expected influence degree over the time span of interest (integral-time maximization problem). Both problems are solved by the greedy algorithm taking advantage of the submodu-

larity of the objective function. We confirmed by applying the proposed method to two real world networks taken from the blog and Wikipedia data that the proposed method can achieve considerable reduction in computation time without degrading the accuracy compared with the naive simulation method as predicted by the theory. Use of the two control strategies contributes to reducing the computational cost by a factor of 50 compared with the naive bond percolation which itself is 2 to 3 orders of magnitudes faster than the naive simulation method. The proposed method can discover nodes that are more influential than the nodes identified by the conventional methods based on the various centrality measures. The results of the two influence maximization problems are totally different in terms of the identified influential nodes and thus it is crucial to choose the right objective function that meets the need for the task. We further found that the pruning is effective when searching for a single influential node, but gradually its overhead surpasses its saving and the burnout is more powerful when searching for multiple influential nodes. Use of both is most effective for the initial few iterations. Thus, we recommend to use both the pruning and the burnout only in the initial few iterations, and stop using the pruning and use the burnout alone in the succeeding iterations in the greedy algorithm. Just as a key task on biology is to find some important groups of genes or proteins by performing biologically plausible simulations over regulatory networks or metabolic pathways, our proposed method can be a core technique for the discovery of influential persons over real social networks.

Acknowledgements. This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

Adar E, Adamic LA (2005) Tracking information epidemics in blogspace. In Skowron A, Agrawal R, Luck M, Yamaguchi T, Morizet-Mahoudeaux P, Liu J, Zhong N (eds). Proceedings of 2005 IEEE/WIC/ACM international conference on Web intelligence, Compiegne, France, September 2005, pp 207–214

Agarwal N and Liu H (2008) Blogosphere: research issues, tools, and applications. SIGKDD Explorations 10(1):18–31

Backstrom L, Dwork C, Kleinberg J (2007) Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In Williamson CL, Zurko ME, Patel-Schneider PF, Shenoy PJ (eds). Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, May 2007, pp 181–190

Chen W, Wang Y, Yang S (2009) Efficient influence maximization in social networks. Elder IV JF, Fogelman-Soulié F, Flach PA, Zaki MJ (eds). Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, Paris, France, June 2009, pp 199–208

Domingos P (2005) Mining social networks for viral marketing. IEEE Intelligent Systems 20(1):80–82

Domingos P, Richardson M (2001) Mining the network value of customers. In Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, August 2001, pp 57–66

Goldenberg J, Libai B, Muller E (2001) Talk of the network: a complex systems look at the underlying process of word-o-mouth. Marketing Letters 12(3):211–223

Grassberger P (1983) On the critical behavior of the general epidemic process and dynamical percolation. Mathematical Bioscience 63(2):157–172

Gruhl D, Guha R, Liben-Nowell D, Tomkins A (2004) Information diffusion through blogspace. In Feldman SI, Uretsky M, Najork M, Wills CE (eds). Proceedings of the 13th international conference on World Wide Web, New York, NY, May 2004, pp 107–117

Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In Getoor L, Senator TE, Domingos P, Faloutsos C (eds). Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining, Washington DC, August 2003, pp 137–146

Kimura M, Saito K, Motoda H (2009a) Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data 3(2):9:1–9:23

Kimura M, Saito K, Motoda H (2009b) Efficient estimation of influence functions for SIS model on social networks. In Boutilier C (ed). Proceedings of the 21st international joint conference on artificial intelligence, Pasadena, CA, July 2009, pp 2046–2051

Kimura M, Saito K, Nakano R (2007) Extracting influential nodes for information diffusion on a social network. In Proceedings of the 22nd AAAI conference on artificial intelligence, Vancouver, British Columbia, Canada, July 2007, pp 1371–1376

Kimura M, Saito K, Nakano R, Motoda H (2010) Extracting influential nodes on a Social Network for information. Data Mining and Knowledge Discovery 20(1):70–97

Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N (2007a) Cost-effective outbreak detection in networks. In Berkhin P, Caruana R, Wu X (eds). Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, San Jose, CA, August 2007, pp 420–429

Leskovec J, McGlohon M, Faloutsos C, Glance N, Hurst M (2007b) Patterns of cascading behavior in large blog graphs. In Proceedings of the Seventh SIAM international conference on data mining, Minneapolis, MN, April 2007, pp 551–556

McCallum A, Corrada-Emmanuel A, Wang X (2005) Topic and role discovery in social networks. In Kaelbling LP, Saffiotti A (eds). Proceedings of the 19th international joint conference on artificial intelligence, Edinburgh, Scotland, UK, July - August 2005, pp 786–791

Mislove A, Marcon M, Gummadi KP, Druschel P, Bhattacharjee B (2007) Measurement and analysis of online social networks. In Dovrolis C, Roughan M (eds). Proceedings of the seventh ACM SIGCOMM conference on internet measurement, San Diego, CA, October 2007, pp 29–42

Muhlestein D, Lim S (2009) Online learning with social computing based interest sharing. Knowledge and Information Systems, Published online: November 2009

Newman MEJ (2001) The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences of the United States of America 98(2):404–409

Newman MEJ (2002) Spread of epidemic disease on networks. Physical Review E 66:016128

Newman MEJ (2003) The structure and function of complex networks. SIAM Review 45(2):167–256

Newman MEJ, Park J (2003) Why social networks are different from other types of networks. Physical Review E 68:036122

Peng W, Li T (2010) Temporal relation co-clustering on directional social network and author-topic evolution. Knowledge and Information Systems, Published online: March 2010

Richardson M, Domingos P (2002) Mining knowledge-sharing sites for viral marketing. In Proceedings of the Eighth ACM SIGKDD international conference on knowledge discovery and data mining, Edmonton, Alberta, Canada, July 2002, pp 61–70

Saito K, Kimura M, Motoda H (2009) Discovering influential nodes for SIS models in social networks. In Gama J, Costa VS, Jorge AM, Brazdil P (eds). Proceedings of the 12th International Conference of Discovery Science, Porto, Portugal, October 2009. Lecture Notes in Computer Science 5808, Springer, pp 302–316

Wasserman S, Faust K (1994) Social network analysis. Cambridge University Press, Cambridge, UK

Watts DJ (2002) A simple model of global cascade on random networks. Proceedings of the National Academy of Sciences of the United States of America 99(9):5766–5771

Watts DJ, Dodds PS (2007) Influence, networks, and public opinion formation. Journal of Consumer Research 34(4):441–458

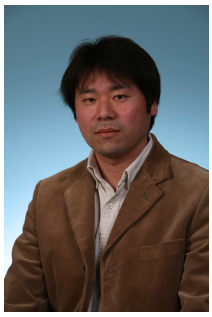
Zhou B, Pei J (2010) The *k*-anonymity and *l*-diversity approaches for privacy preservation in social networks against neighborhood attacks. Knowledge and Information Systems, Published online: June 2010

Zhou D, Ji X, Zha H, Giles CL (2006) Topic evolution and social interactions: how authors effect research. In Yu PS, Tsotras VJ, Fox EA, Liu B (eds). Proceedings of the 2006 ACM CIKM international conference on information and knowledge management, Arlington, VA, November 2006, pp 248–257

Zhuge H, Zhang J (2010) Topological centrality and its applications. Journal of the American Society for Information Science and Technology 61(9):1824-1841

Author Biographies

Kazumi Saito received a BS degree in mathematics from Keio University, Kanagawa, Japan, in 1985, and a PhD in engineering from University of Tokyo, Tokyo, Japan, in 1998. In 1985, he joined the NTT Electrical Communication Laboratories, Kanagawa, Japan. In 1991, he joined the NTT Communication Science Laboratories, Kyoto, Japan. In 2007, he joined the University of Shizuoka, Shizuoka, Japan. He is a professor at the School of Administration and Informatics. From 1991 to 1992, he was a visiting scholar at the University of Ottawa, Ontario, Canada. His current research interests are machine learning and statistical analysis of complex networks. He is a member of the Institute of Electronics, Information, and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ), the Japanese Society of Artificial Intelligence (JSAI), the Japanese Neural Network Society (JNNS).



Masahiro Kimura received his BS, MS, and PhD degrees in mathematics from Osaka University, Osaka, Japan, in 1987, 1989, and 2000, respectively. In April 1989, he joined Nippon Telegraph and Telephone (NTT) Corporation, Tokyo, Japan. He mainly worked at NTT Human Interface Laboratories and NTT Communication Science Laboratories. In April 2005, he joined Ryukoku University, Kyoto, Japan. Currently, he serves as a professor of the Department of Electronics and Informatics. His research interests include complex networks science, data mining, and machine learning. He is a member of the Japanese Society for Artificial Intelligence (JSAI), the Mathematical Society of Japan (MSJ), the Japan Society for Industrial and Applied Mathematics (JSIAM), the Japanese Neural Networks Society (JNNS), and the Institute of Electronics, Information and Communication Engineers (IEICE).

Kouzou Ohara received the Master of Engineering degree from Osaka University, Osaka, Japan in 1995. He also received the Ph. D. degree in engineering from Osaka University in 2002. He is currently an Associate Professor in the department of Integrated Information Technology at the college of Science and Engineering of Aoyama Gakuin University. His research interests include machine learning, data mining, social network analysis, and personalization of intelligent systems. He is a member of the Institute of Electrical and Electronics Engineers (IEEE), the Association for the Advancement of Artificial Intelligence (AAAI), the Institute of Electronics, Information, and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ) and the Japanese Society of Artificial Intelligence (JSAI).

Hiroshi Motoda is a professor emeritus of Osaka University and a scientific advisor of AFOSR/AOARD (Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, US Air Force Research Laboratory). His research interests include information diffusion in social network, data mining, machine learning, knowledge acquisition, scientific knowledge discovery and artificial intelligence in general. He received his Bs, Ms and PhD degrees all in nuclear engineering from the University of Tokyo. He is a member of the steering committee of PAKDD, PRICAI, DS and ACML. He received the best paper awards from Atomic Energy Society of Japan (1977, 1984) and from Japanese Society of Artificial Intelligence (1989, 1992, 2001), the outstanding achievement awards from JSAI (2000) and Okawa Publication Prize from Okawa Foundation (2007).

Correspondence and offprint requests to: Masahiro Kimura, Department of Electronics and Informatics, Ryukoku University, Otsu 520-2194, Japan. Email: kimura@rins.ryukoku.ac.jp

Estimating Diffusion Probability Changes for AsIC-SIS Model from Information Diffusion Results

Akihiro Koide J11103@U-SHIZUOKA-KEN.AC.JP
Graduate School of Management and Information of Innovation
University of Shizuoka

Kazumi Saito K-SAITO@U-SHIZUOKA-KEN.AC.JP
School of Management and Information
University of Shizuoka

Kouzou Ohara OHARA@IT.AOYAMA.AC.JP
Department of Integrated Information Technology
Aoyama Gakuin University

Masahiro Kimura KIMURA@RINS.RYUKOKU.AC.JP
Department of Electronics and Informatics
Ryukoku University

Hiroshi Motoda MOTODA@AR.SANKEN.OSAKA-U.AC.JP
Institute of Scientific and Industrial Research
Osaka University

Editor:

Abstract

We address the problem of estimating changes in diffusion probability over a social network from the observed information diffusion results, which is possibly caused by an unknown external situation change. For this problem, we focused on the asynchronous independent cascade (AsIC) model in the SIS (Susceptible/Infected/Susceptible) setting in order to meet more realistic situations such as communication in a blogosphere. This model is referred to as the AsIC-SIS model. We assume that the diffusion parameter changes are approximated by a series of step functions, and their changes are reflected in the observed diffusion results. Thus, the problem is reduced to detecting how many step functions are needed, where in time each one starts and how long it lasts, and what the hight of each one is. The method employs the derivative of the likelihood function of the observed data that are assumed to be generated from the AsIC-SIS model, adopts a divide-and-conquer type greedy recursive partitioning, and utilizes an MDL model selection measure to determine the adequate number of step functions. The results obtained using real world network structures confirmed that the method works well as intended. The MDL criterion is useful to avoid overfitting, and the found pattern is not necessarily the same in terms of the number of step functions as the one assumed to be true, but the error is always reduced to a small value.

Keywords: pattern change detection, information diffusion, parameter learning, social networks

1. Introduction

Recent technological innovation in the web such as blogosphere and knowledge/media-sharing sites is remarkable, which has made it possible to form various kinds of large social networks, through which behaviors, ideas, rumors and opinions can spread, and our behavioral patterns are to a con-

siderable degree affected by the interaction with these networks and substantial attention has been directed to investigating the spread of information in these networks (Newman et al., 2002; Newman, 2003; Gruhl et al., 2004; Domingos, 2005; Leskovec et al., 2006; Crandall et al., 2008; Wu and Huberman, 2008).

These studies have shown that it is important to consider the diffusion mechanism explicitly and the measures based on network structure alone, *i.e.*, various centrality measures such as degree, betweenness, closeness, etc., are not enough to identify the important nodes (Kimura et al., 2009a, 2010a). Information diffusion is modeled typically by probabilistic models. Most representative and fundamental ones are independent cascade (IC) model (Goldenberg et al., 2001; Kempe et al., 2003), linear threshold (LT) model (Watts, 2002; Watts and Dodds, 2007) and their extensions that include incorporating asynchronous time delay (Saito et al., 2009b, 2010a). In the IC model the information sender (a node) tries to push the information to the neighboring receivers (child nodes) in a probabilistic way, whereas in the LT model the information receiver (a node) tries to pull the information from the neighoring senders (parents nodes) in a probabilistic way. These models place the constraint that a node is given a single chance to activate the other node, *i.e.*, the same node is not activated multiple times. This setting is called SIR (Susceptible/Infectious/Recovered) in analogy with epidemic disease. Explicit use of these models to solve such problems as the *influence maximization problem* (Kempe et al., 2003; Kimura et al., 2010a) and the *contamination minimization problem* (Kimura et al., 2009a) clearly shows the advantage of the model. They showed that the identified influential nodes and links are considerably different from the ones identified by the standard centrality measures. The SIR setting is simple, but does not model well such communication as in a blogosphere where the same person can post on the same topic multiple times. The SIS (Susceptible/Infectiouse/Susceptible) setting is better suited to this situation, where a node is allowed to activate the other nodes multiple times, *i.e.*, the same node is activated multiple times (Kimura et al., 2009b).

What is common to all the above models is that they are all probabilistic models and have parameters to characterize the information diffusion, and these parameters are assumed to be stationary, *i.e.*, they do not change over time. Evidently, the parameters must be known in advance for the model to be usable for analysis, but it is generally difficult to determine the values of these parameters theoretically. Therefore, attempts have been made to learn these parameter values by the observed information diffusion sequence data (Saito et al., 2009a,b, 2010a,b; Gomez-Rodriguez et al., 2010; Myers and Leskovec; Kimura et al., 2010b). In essence the likelihood of generating the observed data by the model employed is first derived, and then the parameter values are determined such that the likelihood is maximized. In particular, Myers and Leskovec showed that for a certain class of diffusion models, the problem can effectively be transformed to a convex programming for which a global solution is guaranteed.

This paper also deals with a parameter learning problem, but addresses a different aspect of information diffusion. We do not assume that the parameter values are stationary, but allow that they change over time. They may change abruptly or gradually depending on the cause of changes which we do not know. Ideally we intend to be able to deal with any shape of changes over time. However, in this paper, we limit the change pattern to those that can be approximated by a series of step functions, and further assume that the change takes place uniformly in space, *i.e.*, the parameters of all nodes change in the same way. We use AsIC-SIS, Asynchronous Independent Cascade model in SIS setting. This is a model in which the original discrete time step IC-SIR model is extended to continuous time model allowing asynchronous time delay (Saito et al., 2009b, 2010a) as well as

allowing multiple activations of the same nodes. We learn the parameter values from an observed sequence of information diffusion under AsIC-SIS model setting, *i.e.*, the problem is reduded to detecting how many step functions are needed, where in time each one starts and how long it lasts, and what the hight of each one is. This is viewed as a generalization of our previous work (Ohara et al., 2011) in which we used the AsIC-SIR model, limited the change pattern to be a single rect-linear shape, and devised an efficient algorithm which searches the optimal window. However, this algorithm works only to this restricted type of the problem.

We extended the parameter optimization algorithm that was developed in Saito et al. (2009b); Kimura et al. (2010b), *i.e.*, the EM-like algorithm for the AsIC-SIR model that iteratively updates the values to maximize the model's likelihood of generating the observed data sequences, to AsIC-SIS. The core part of this paper is how to efficiently search the change pattern. We employed the idea of using the first order derivative of the likelihood with respect to the parameters (Ohara et al., 2011), and newly developed an efficent algorithm that uses a divide-and-conquer type greedy recursive partitioning as a search strategy and an MDL model selection measure as a stopping criterion to determine the most adequate number of step functions. We tested our algorithm to artificially generated change patterns using four real world network structures. The results obtained confirmed that the method works well as intended. The algoritm is efficent because it needs to do expensive parameter optimization only once for each partitioning (which is not that many in many cases). The MDL criterion is useful to avoid overfitting. In many cases it identifies the correct number of step functions, but in some cases the found pattern is not necessarily the same in terms of the number of step functions, but the error is always reduced to a small value.

The paper is organized as follows. After very briefly introducing the AsIC-SIS model in Section 2, we define the problem in Section 3 and derive the liklihood function in Section 4, which is the objective function to be maximized. The parameter estimation algorithm is summarized in Appendix. We then describe how we efficiently search for the change pattern in Section 5 together with the restricted search method used for comparative study. The experimental results are reported in Section 6. We end this paper by summarizing the main result in Section 7.

2. Information Diffusion Model

An SIS model for the spread of a disease is based on the cycle of disease in a host. A person is first *susceptible* to the disease, and becomes *infected* with some probability and time-delay if he or she has contact with an infected person. The infected person becomes susceptible to the disease again without moving to the immune state. We consider an asynchronous-time SIS model for information diffusion on a network. In this context, infected nodes mean that the nodes have adopted the information, and we call these infected nodes *active* nodes. This can be mapped to realistic situations such as communication in a blogsphere. A typical example would be the following propagation phenomenon of a topic in the blogsphere: A blogger who has not yet posted a message about a certain topic becomes interested in the topic by reading the blog of his or her friend, and posts a message about it with some time-delay from the friend's posting time, *i.e.*, becoming infected (activated) with some time-delay. Right after posting the message, the same blogger can read any other blogs of his or her friends, *i.e.*, becoming susceptible again. The same blogger reads a new message about the topic posted by some other friend, and may post another message, *i.e.*, becoming infected again. This process is repeated.

Let $G = (V, E)$ be a directed network, where V and E stand for the sets of all the nodes and (directed) links, respectively. Here, note that E is a subset of $V \times V$. For any $v \in V$, the set of all the nodes that have links from v (child nodes) is denoted by $F(v) = \{u \in V;\ (v, u) \in E\}$, and the set of all the nodes that have links to v (parent nodes) is denoted by $B(v) = \{u \in V;\ (u, v) \in E\}$. We define the AsIC-SIS model for information diffusion on G. In the model, the diffusion process unfolds in continuous-time $t \geq 0$, and it is assumed that the state of a node is either active or inactive. For every link $(u, v) \in E$, we specify a real value $p_{u,v}$ with $0 < p_{u,v} < 1$ in advance. Here, $p_{u,v}$ is referred to as the *diffusion probability* through link (u, v). Given an initial active node v and a time span T, the diffusion process proceeds in the following way. Suppose that node u becomes active at time t ($< T$). Then, node u attempts to activate every $v \in F(u)$, and succeeds with probability $p_{u,v}$. If node u succeeds, then node v will become active at time $t + \delta$. We assume that a delay-time δ is chosen from some probability distribution, and we used the exponential distribution with parameter $r_{u,v}$ for the sake of convenience, but of course other distributions such as power-law and Weibull can be employed. Suppose that u, one of the parent nodes of v, succeeds to activate v at some time after some delay. In our SIS model, when some other parent node also succeeds to activate v before it gets activated by u, we assume that v's activation time is overridden by the one earliest possible. On the other hand, node u gets back inactive right after time t (the time it gets activated) and it can only be reactivated by those parent nodes that have become active after time t[1]. The process terminates if the current time reaches the time limit T.

The AsIC-SIS model is the SIS version of the asynchronous independent cascade (AsIC) model proposed by Saito et al. (2009b) that is an extension of the independent cascade (IC) model studied by Kempe et al. (2003). As mentioned earlier, the AsIC-SIS model was extended to meet more realistic situations.

3. Problem Definition

We address the *problem of estimating diffusion probability changes*. In this problem, we assume that some changes have happened in the way the information diffuses, and we observe the diffusion results of a certain topic in which the changes are embedded, and consider estimating the diffusion probability as a function with respect to time t.

An information diffusion result generated by the AsIC-SIS model is represented as a set of pairs of active nodes and their activation times; *i.e.*, $\{\cdots, (v(\eta),\ t_{v(\eta)}), \cdots\}$, where $v(\eta)$ indicates v's η-th activation. We consider a diffusion result $\mathcal{D}(0, T)$, where the initial activation time is set to 0 and the final observation time is denoted by T. Since we employ only a single diffusion result $\mathcal{D}(0, T)$, we place a constraint that $p_{u,v}$ and $r_{u,v}$ do not depend on link (u, v), *i.e.*, $p_{u,v} = p$, $r_{u,v} = r$ $(\forall (u, v) \in E)$, which should be acceptable noting that we can naturally assume that people behave quite similarly when talking about the same topic. In fact, our previous experiments (Saito et al., 2009b, 2010a,b) give some evidences which support the validity of this constraint.

Let $p(t)$ be a function of diffusion probability with respect to time t. Here we assume that $p(t)$ is reasonably approximated by combining a number of step functions, *i.e.*,

$$p(t) \ = \ p_{i-1} \quad \text{if} \ \ t \in [t_{i-1}, t_i), \quad i \in \{1, \cdots, K+1\}, \tag{1}$$

1. In theory we can go back to all the past time points at which the parents of u got activated multiple times in the past, but this is unrealistic and we thought it natural to limit the parents only to those that got activated after time t.

where $t_0 = 0 < \cdots < t_i < \cdots t_{K+1} = T$ and K stands for the number of change points. Here we assume for simplicity that the time-delay parameter r does not change and takes the same value for the entire period $[0, T)$. Then, the diffusion probability estimation problem is reduced to detecting the change points $\{t_1, \cdots, t_K\}$ and estimating the associated diffusion probabilities $\{p_0, \cdots, p_K\}$ from the observed diffusion result $\mathcal{D}(0, T)$. For a given integer K, we define the *change point vector* $\boldsymbol{t}_K$ and the *diffusion-probability vector* $\boldsymbol{p}_K$ by $\boldsymbol{t}_K = (t_1, \cdots, t_K)$ and $\boldsymbol{p}_K = (p_0, \cdots, p_K)$, respectively.

4. Model parameter learning

We describe the framework of model parameter learning as a likelihood maximization problem for the AsIC-SIS model.

First, we consider estimating the values of diffusion probability p and time-delay parameter r from an observed diffusion result $\mathcal{D}(0, T) = \{\cdots, (v(\eta),\ t_{v(\eta)}), \cdots\}$ when there is no change point. Recall that the initial activation time is set to 0 and the final observation time is denoted by T. Let $\mathcal{D}$ be the set of all the activated nodes in $\mathcal{D}(0, T)$, *i.e.*, $\mathcal{D} = \{v(\eta) \in V;\ (v(\eta),\ t_{v(\eta)}) \in \mathcal{D}(0, T)\}$. For each node $v(\eta) \in \mathcal{D}$, let $\mathcal{AP}_{v(\eta)}$ be the set of its parent nodes that had a chance to activate it, *i.e.*,

$$\mathcal{AP}_{v(\eta)} = \{u(\zeta);\ u \in B(v),\ (u(\zeta),\ t_{u(\zeta)}) \in \mathcal{D}(0, T),\ t_{v(\eta-1)} < t_{u(\zeta)} < t_{v(\eta)}\},$$

and $\mathcal{NC}_{v(\eta)}$ be the set of its child nodes that was not activated by a node $v(\eta)$ within $(t_{v(\eta)}, T)$, *i.e.*,

$$\mathcal{NC}_{v(\eta)} = \{z \in F(v);\ \neg\exists\, z(\xi),\ \text{s.t. } (z(\xi),\ t_{z(\xi)}) \in \mathcal{D}(0, T),\ t_{v(\eta)} < t_{z(\xi)} < T)\}.$$

Note that from the observed diffusion result, we know that a node v at the η-th activation did not succeed to activate any child node in $\mathcal{NC}_{v(\eta)}$ within the time limit T, and we use this fact for our parameter estimation in order to improve its performance.

Let $\mathcal{X}_{u(\zeta),v(\eta)}(p, r)$ denote the probability density that a node $u(\zeta) \in \mathcal{AP}_{v(\eta)}$ activates the node $v(\eta)$ at time $t_{v(\eta)}$, that is,

$$\mathcal{X}_{u(\zeta),v(\eta)}(p, r) \quad = \quad p\, r \exp(-r(t_{v(\eta)} - t_{u(\zeta)})). \tag{2}$$

Let $\mathcal{Y}_{u(\zeta),v(\eta)}(p, r)$ denote the probability that the node $v(\eta)$ is not activated by a node $u(\zeta) \in \mathcal{AP}_{v(\eta)}$ within the time-period $(t_{u(\zeta)},\ t_{v(\eta)})$, that is,

$$\begin{aligned} \mathcal{Y}_{u(\zeta),v(\eta)}(p, r) &= 1 - p \int_{t_{u(\zeta)}}^{t_{v(\eta)}} r \exp(-r(t - t_{u(\zeta)}))dt \\ &= p \exp(-r(t_{v(\eta)} - t_{u(\zeta)})) + (1 - p). \end{aligned} \tag{3}$$

By using Eqs. (2) and (3), we can obtain the probability density $\phi_{v(\eta)}(p, r)$ that some node $u(\zeta) \in \mathcal{AP}_{v(\eta)}$ succeeds to activate a node $v(\eta)$ at a time $t_{v(\eta)}$,

$$\phi_{v(\eta)}(p, r) \quad = \quad \sum_{u(\zeta) \in \mathcal{AP}_{v(\eta)}} \mathcal{X}_{u(\zeta),v(\eta)}(p, r) \left(\prod_{z(\xi) \in \mathcal{AP}_{v(\eta)} \setminus \{u(\zeta)\}} \mathcal{Y}_{z(\xi),v(\eta)}(p, r) \right). \tag{4}$$

and the probability $\psi_{v(\eta)}(p, r)$ that a node $v(\eta)$ cannot activate any node $z \in \mathcal{NC}_{v(\eta)}$ within $(t_{v(\eta)}, T)$,

$$\psi_{v(\eta)}(p, r) \quad = \quad \left(p \exp(-r(T - t_{v(\eta)})) + (1 - p)\right)^{|\mathcal{NC}_{v(\eta)}|}. \tag{5}$$

Then, from Eqs. (4) and (5), the following log likelihood function $\mathcal{L}(p, r; \mathcal{D}(0, T))$ can be obtained for observed data $\mathcal{D}(0, T)$

$$\mathcal{L}(p, r; \mathcal{D}(0, T)) \quad = \quad \sum_{v(\eta) \in \mathcal{D}} \Big(\log \phi_{v(\eta)}(p, r) + \log \psi_{v(\eta)}(p, r) \Big). \tag{6}$$

The values of parameters p and r can be stably obtained by maximizing Eq. (6) using an EM-like algorithm. (see Appendix A for more details).

Now, we assume that there exist change points specified by the change point vector $\boldsymbol{t}_K$ and the associated diffusion-probability vector $\boldsymbol{p}_K$. For any $v(\eta) \in \mathcal{D}(0, T)$, let $\phi_{v(\eta)}(\boldsymbol{p}_K, r; \boldsymbol{t}_K)$ be the probability density that some node $u(\zeta) \in \mathcal{AP}_{v(\eta)}$ succeeds to activate a node $v(\eta)$ at time $t_{v(\eta)}$, *i.e.*,

$$\phi_{v(\eta)}(\boldsymbol{p}_K, r; \boldsymbol{t}_K) \quad = \quad \sum_{u(\zeta) \in \mathcal{AP}_{v(\eta)}} \mathcal{X}_{u,v}(p(t_{u(\zeta)}), r) \prod_{z(\xi) \in \mathcal{AP}_{v(\eta)} \setminus \{u(\zeta)\}} \mathcal{Y}_{z,v}(p(t_{z(\xi)}), r) \tag{7}$$

and $\psi_{v(\eta)}(p(t_{v(\eta)}), r; \boldsymbol{t}_K)$ be the probability that a node $v(\eta)$ cannot activate any node $z \in \mathcal{NC}_{v(\eta)}$ within $(t_{v(\eta)}, T]$, *i.e.*,

$$\psi_{v(\eta)}(p(t_{v(\eta)}), r; \boldsymbol{t}_K) \quad = \quad \Big(p(t_{v(\eta)}) \exp(-r(T - t_{v(\eta)})) + (1 - p(t_{v(\eta)})) \Big)^{|\mathcal{NC}_{v(\eta)}|}. \tag{8}$$

Using Eqs. (7) and (8), we can define the following objective function $\mathcal{L}(\boldsymbol{p}_K, r; \mathcal{D}(0, T), \boldsymbol{t}_K)$.

$$\mathcal{L}(\boldsymbol{p}_K, r; \mathcal{D}(0, T), \boldsymbol{t}_K) \quad = \quad \sum_{v(\eta) \in \mathcal{D}} \Big(\log \phi_{v}(\eta)(\boldsymbol{p}_K, r; \boldsymbol{t}_K) + \log \psi_{v(\eta)}(p(t_{v(\eta)}), r; \boldsymbol{t}_K) \Big). \tag{9}$$

Clearly, $\mathcal{L}(\boldsymbol{p}_K, r; \mathcal{D}(0, T), \boldsymbol{t}_K)$ is expected to be maximized by setting $\boldsymbol{t}_K$ to the true change points vector $\boldsymbol{t}^*_K = (t^*_1, \cdots, t^*_K)$ if a substantial amount of data $\mathcal{D}(0, T)$ is available. Thus, our diffusion probability estimation problem is formalized as the following maximization problem:

$$\hat{\boldsymbol{t}_K} \quad = \quad \arg\max_{\boldsymbol{t}_K} \mathcal{L}(\hat{\boldsymbol{p}}_K(\boldsymbol{t}_K), \hat{r}(\boldsymbol{t}_K); \mathcal{D}(0, T), \boldsymbol{t}_K), \tag{10}$$

where $\hat{\boldsymbol{p}}_K(\boldsymbol{t}_K)$ and $\hat{r}(\boldsymbol{t}_K)$ denote the maximum likelihood estimators for a given $\boldsymbol{t}_K$.

5. Estimation Methods

For a given number of change points, K, in order to obtain the optimal change point vector $\hat{\boldsymbol{t}}_K$ according to Eq. (10), we need to prepare a reasonable set of candidate change poits, denoted by $\mathcal{T}$. One way of doing so is to construct $\mathcal{T}$ by considering all of the observed activation time points.

$$\mathcal{T} = \{t_{v(\eta)};\ (v(\eta),\ t_{v(\eta)}) \in \mathcal{D}(0, T)\} \cup \{T\} = \{\tau_0, \tau_1, \cdots, \tau_N\}, \quad (0 = \tau_0 < \tau_1 < \cdots < \tau_N = T).$$

Here N is equal to the number of activated nodes in a information diffusion result, *i.e.*, $N = |\mathcal{D}(0, T)|$. Hereafter, we denote the model parameter vector by $\boldsymbol{\theta}_K$; *i.e.*, $\boldsymbol{\theta}_K = (\boldsymbol{p}_K, r)$ for the AsIC-SIS model.

5.1 Proposed Method

Our proposed method employs a greedy strategy. Clearly, we can obtain the parameter vector $\boldsymbol{\theta}_0$ from the original objective function of Eq. (6). Now, under the condition that we have obtained the

K chage point(s), we consider selecting the next $(K+1)$-th change point. Of course, we can obtain the maximum likelihood estimators, $\hat{\boldsymbol{\theta}}_K$, from the extended objective function of Eq. (9). Then, we focus on the first-order partial derivative of the objective function $\mathcal{L}(\hat{\boldsymbol{\theta}}_K; \mathcal{D}(0,T))$ with respect to a new parameter $p_{v(\eta)}$ introduced by considering as if each node $v \in V$ has an individual diffusion probability $p_{v(\eta)}$ at each activation time $t_{v(\eta)}$. Note that under this situation, by posing the restriction of parameter sharing setting, defined by $p_{v(\eta)} = p_i$ if $t_{v(\eta)} \in [t_i, t_{i+1})$, we obtain each maximum likelihood estimator by $\hat{p}_{v(\eta)} = \hat{p}_i$. Thus, from the optimal necessary condition of the maximum likelihood estimation, we have

$$0 = \frac{\partial \mathcal{L}(\hat{\boldsymbol{\theta}}_K; \mathcal{D}(0,T))}{\partial p_i} = \sum_{t_{v(\eta)} \in [t_i, t_{i+1})} \frac{\partial \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}_K; \mathcal{D}(0,T))}{\partial p_{v(\eta)}}. \tag{11}$$

Now we assume that there exists an undetected change point $t_j \in [t_i, t_{i+1})$. Then the estimated parameter $\hat{p}_i$ for the time span $[t_i, t_{i+1})$ is nothing but a compromised value between diffusion probabilities of $[t_i, t_j)$ and $[t_j, t_{i+1})$. Thus, we can expect that the following relation holds for the product of the partial derivatives between many pairs of $p_{u(\zeta)}$ and $p_{v(\eta)}$ if both $t_{u(\zeta)}$ and $t_{v(\eta)}$ are included in either before the change point $[t_i, t_j)$ or after the change point $[t_j, t_{i+1})$.

$$\frac{\partial \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}_K; \mathcal{D}(0,T))}{\partial p_{u(\zeta)}} \frac{\partial \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}_K; \mathcal{D}(0,T))}{\partial p_{v(\eta)}} > 0 \tag{12}$$

Here, we consider the following partial sum for the derivatives:

$$g(\tau_n) = \sum_{t_{v(\eta)} < \tau_n} \frac{\partial \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}_K; \mathcal{D}(0,T))}{\partial p_{v(\eta)}}, \quad n = 1, \cdots, N, \tag{13}$$

where $g(\tau_n) = 0$ if $\tau_n = t_i$. By Eqs. (11) to (13), we can expect that $|g(n)|$ is locally maximized at each undetected change point $\tau_n = t_j$. This is because the sign of the product of the partial derivatives $\partial \mathcal{L}(\hat{\boldsymbol{\theta}}_K; \mathcal{D}(0,T))/\partial p_{u(\zeta)}$ and $\partial \mathcal{L}(\hat{\boldsymbol{\theta}}_K; \mathcal{D}(0,T))/\partial p_{v(\eta)}$ changes at the boundaries of the undetected change points $\{t_j\}$. Therefore, we propose the method of detecting the next change point by

$$\hat{\tau}_n = \arg\max_{\tau_n \in \mathcal{T}} |g(\tau_n)|. \tag{14}$$

Here note that we can incrementally calculate $g(\tau_n)$. More specifically, we can obtain the following formula by $t_{v(\eta)} = \tau_{n+1}$:

$$g(\tau_{n+1}) = g(\tau_n) + \frac{\partial \mathcal{L}(\hat{\boldsymbol{\theta}}_K; \mathcal{D}(0,T))}{\partial p_{v(\eta)}} \tag{15}$$

for any $\tau_n, \tau_{n+1} \in \mathcal{T}$.

Thus far, we assumed that the number of change points, K, is known. However, since this assumption does not hold in many applications, we need to obtain an adequate K from a given diffusion result. For this purpose, we can utilize some statistical measure such as MDL (Minimum Description Length). Note that due to a time-series nature of our observation data, we cannot

straightforwardly apply a resampling technique such as k-fold cross-validation for this model selection. In our experiments, we employed the following MDL value.

$$MDL(\boldsymbol{\theta}_K) \quad = \quad -\mathcal{L}(\hat{\boldsymbol{\theta}}_K; \mathcal{D}(0,T)) + (K+1)\log M, \quad M \; = \; \sum_{v(\eta)\in\mathcal{D}} |F(v(\eta))|\,, \tag{16}$$

where $K+1$ and M correspond to the number of parameters and the number of coin-flips performed by the AsIC-SIS model, respectively. Note that we regard M as the number of samples for our learning. Then we can summarize our proposed method below.

1. Set $K = 0$ and $\boldsymbol{t}_0$ to an empty list, and intialize $\boldsymbol{\theta}_0$ adequately.
2. Maximize $\mathcal{L}(\boldsymbol{\theta}_K; \mathcal{D}(0,T))$ by using the parameter estimation method, and calculate $MDL(\boldsymbol{\theta}_K)$.
3. If $K > 0$ and $MDL(\boldsymbol{\theta}_K) > MDL(\boldsymbol{\theta}_{K-1})$, output $\boldsymbol{t}_{K-1}$ and $\boldsymbol{\theta}_{K-1}$.
4. Detect the change point $\hat{\tau}_n$ by Eq. (14), construct $\boldsymbol{t}_{K+1}$ by adding $\hat{\tau}_n$ to $\boldsymbol{t}_K$, set $K = K+1$, and return to step 2.

Here note that the proposed method requires likelihood maximization by using the parameter estimation method only $(K+1)$ times.

5.2 Comparison Method

As mentioned earlier, we have already proposed a hot span detection method for the AsIC model in the SIR (Susceptible/Infected/Recover) setting, although this method is only applicable to a restricted form of the change pattern expressed by a pair of $\boldsymbol{t}_2 = (t_1, t_2)$ and $\boldsymbol{p}_2 = (p_0, p_1, p_0)$ (Ohara et al., 2011). The results reported are good. Thus, we extend this method to the SIS (Susceptible/Infected/Susceptible) setting, and use the extended method for performance comparison, knowing that the method is intended to a single rect-linear pattern change. In what follows, we outline this method.

The comparison method also utilizes a modified version of Eq. (13) as the measure of interval selection, expressed by

$$[\hat{\tau}_m, \hat{\tau}_n) \quad = \quad \arg\max_{\tau_m,\tau_n\in\mathcal{T}} \left| \sum_{t_{v(\eta)}\in[\tau_m,\tau_n)} \frac{\partial\mathcal{L}(\hat{\boldsymbol{\theta}}_K; \mathcal{D}(0,T))}{\partial p_{v(\eta)}} \right|. \tag{17}$$

However, this method can be extremely inefficient when the number of candidate time points N is large. Thus, in order to make it work with a reasonable computational cost, we consider restricting the number of candidate time points to a smaller value, denoted by J, *i.e.*, we construct $\mathcal{T}_J$ $(\subset \mathcal{T})$ by randomly selecting J points from $\mathcal{T}$; then we construct a restricted set of candidate spans by

$$\mathcal{H}_J \quad = \quad \{S = [\tau_i, \tau_j);\ \tau_i < \tau_j, \tau_i \in \mathcal{T}_J, \tau_j \in \mathcal{T}_J\}.$$

Note that $|\mathcal{H}_J| = J(J-1)/2$, which is large when J is large.

6. Experimental Evaluation

We experimentally evaluated, given an observed diffusion result, how accurately the proposed method can estimate diffusion probability changes underlying it by investigating the difference between the estimated change pattern and the one that is assumed true using four real world networks.

6.1 Datasets

Here we adopted four large networks in the real world, all of which are bidirectional. The first one is a trackback network of Japanese blogs used in Kimura et al. (2009a), where there are $12,047$ nodes and $79,920$ directed links (the blog network). The second one is a network representing the co-occurrence relation extracted from the "list of people" within Japanese Wikipedia that is used in Kimura et al. (2008), which has $9,481$ nodes and $245,044$ directed links (the Wikipedia network). The third one is a network derived from the Enron Email Dataset (Klimt and Yang, 2004) where the sender and the recipient extracted from the dataset were linked if they had bidirectional communications. It contains $4,254$ nodes and $44,314$ directed links (the Enron network). The last one is a coauthorship network employed in Palla et al. (2005). It has $12,357$ nodes and $38,896$ directed links (the coauthorship network).

6.2 Experimental Setting

We generated diffusion results using the AsIC-SIS model for each of the above networks under the following setting. We considered $p = 1/\bar{d}$ as the base value of the diffusion probability of each link in a network, where $\bar{d}$ is the mean out-degree of the network. For an arbitrary node in the network, the expected number of its children that it succeeds to activate is approximately one at least at an early phase of the information diffusion for this base value. If the diffusion probability is much smaller than the base value, the diffusion process could terminate soon resulting in only few active nodes on the average. If it is much larger, the information rapidly spreads out the entire network and the majority of nodes could be active at any time point in the process, which would also be unrealistic. As a result, too little or too much amount of information diffusion is inappropriate to our aim of investigating the diffusion probability change estimation. Thus, we set the initial diffusion probability, p_0, to be a value slightly smaller than the base value, which is 0.10 for the blog network, 0.02 for the Wikipedia network, 0.05 for the Enron network, and 0.20 for the Coauthorship network, respectively. We considered two kinds of change pattern: one is a rect-linear pattern that has two change points, which is the same as the one used in Ohara et al. (2011) and can be regarded as the most fundamental; and the other is a two-step pattern having three change points, which represents a situation where an event that caused an increase in the diffusion probability of a certain topic occurred, followed by an even bigger event that futher increased the probability, and then the probability returned back to the normal value due to the cease of the event. As for the former pattern, we set the diffusion probability during the second period, p_1, to be three times as large as p_0, and the probability during the third period, p_2, to be the same as p_0. Table 1 summarizes the diffusion probability $\boldsymbol{p}_2^*$ that is assumed true. For all the networks we used the same $\boldsymbol{t}_2^* = (10, 15)$ as the change point vector that is assmued true and $T = 20$ as the final observation time. As for the latter pattern, we set the second and the third diffusion probability, p_1 and p_2, to be twice and three times as large as p_0, respectively, and the last one, p_3 to be the same as p_0. Table 2 summarizes the diffusion probability $\boldsymbol{p}_3^*$ that is assumed true. We used $\boldsymbol{t}_3^* = (10, 15, 20)$ and $T = 25$ for all the networks. As we mentioned in Section 3, we assumed that the time delay parameter does not change, and fixed its value to be 1 ($r = 1$) for every network as changing r works only for scaling the time axis of the diffusion results. In all we generated 100 information diffusion results for each pattern, using the above parameter values, each starting from a randomly selected initial active node for each network.

Table 1: The diffusion probability $\boldsymbol{p}_2^*$ that is assumed true for each of the networks .

diffusion probability ($\boldsymbol{p}_2^*$)	Blog	Wikipedia	Enron	Coauthorship
p_0	0.10	0.02	0.05	0.20
p_1	0.30	0.06	0.15	0.60
p_2	0.10	0.02	0.05	0.20

Table 2: The diffusion probability $\boldsymbol{p}_3^*$ that is assumed true for each of the networks .

diffusion probability ($\boldsymbol{p}_3^*$)	Blog	Wikipedia	Enron	Coauthorship
p_0	0.10	0.02	0.05	0.20
p_1	0.20	0.04	0.10	0.40
p_2	0.30	0.06	0.15	0.60
p_3	0.10	0.02	0.05	0.20

The initial values for p_0 and r were set to a reasonably small random value and a random value around 1, respectively. The termination condition of our parameter learning was as follws:

$$\max_{\theta_i \in \boldsymbol{\theta}_K} |\partial \mathcal{L}(\boldsymbol{\theta}_K; \mathcal{D}(0,T))/\partial \theta_i| < 10^{-4}.$$

We then estimated both the change point vector $\hat{\boldsymbol{t}}_K$ and the model parameter vector $\hat{\boldsymbol{\theta}}_K$, and evaluated their accuracy by integrating the absolute error of the estimated diffusion probability with respect to time t, *i.e.*,

$$\mathcal{E} = \int_0^T |p^*(t) - \hat{p}(t; \hat{\boldsymbol{t}}_K, \hat{\boldsymbol{\theta}}_K)| dt,$$

where $p^*(t)$ is the diffusion probability that is assumed true at time t and $\hat{p}(t; \hat{\boldsymbol{t}}_K, \hat{\boldsymbol{\theta}}_K)$ is its estimation. The estimation with a smaller $\mathcal{E}$ is a better approximation of the true change pattern. In this regards it is not essential that the estimated number of change points, $\hat{K}$, is identical to K^*, the number of change points used to generate the diffusion result. What matters is how close is the estimated pattern as a whole to the true patteren. In fact, K^* is unknown in reality.

6.3 Experimental Results

Table 3 summarizes the results for the first (rect-linear) change pattern, where the integrated estimation errors are the average over independent 100 trials for distinct 100 diffusion results. Here we executed our method until $K = 10$ ignoring the stopping condition at Step 3 of the algorithm shown in Section 5.1, and investigated how the estimation error $\mathcal{E}$ changes over K. The value in the parenthes is the number of trials where the MDL value defined by Eq. (16) took the minimal at that K, which is what the proposed method outputs as the optimal pattern. The row indicated by "MDL estimation" contains the averaged integral error of such optimal patterns. In addition, we showed the estimation error for the comparison method described in Section 5.2 in the row indicated by "Comparison method" as a reference value for evaluation, where J was set to $1,000$.

From these results, we see that the esitmation error drastically drops down at $K = 2$ $(= K^*)$ for every network, which means that the proposed method succeeded in detecting the correct change points and estimating the diffusion probabilities in good accuracy. In fact, the errors of the optimal

Table 3: Integral error $\mathcal{E}$ of the proposed method averaged over 100 trials to estimate a rect-linear change pattern (the value in parentheses is the number of trials where the obtained pattern took the minimal MDL value at K).

♯change points (K)	Blog	Wikipedia	Enron	Coauthorship
0	1.296 (0)	0.273 (5)	0.692 (0)	3.494 (0)
1	1.610 (0)	0.348 (0)	0.575 (0)	3.575 (0)
2 ($= K^*$)	0.126 (64)	0.150 (25)	0.025 (74)	0.614 (7)
3	0.130 (12)	0.108 (41)	0.029 (12)	0.176 (31)
4	0.134 (16)	0.099 (10)	0.032 (6)	0.162 (29)
5	0.136 (4)	0.084 (7)	0.036 (4)	0.156 (12)
6	0.139 (1)	0.081 (4)	0.037 (4)	0.153 (6)
7	0.139 (2)	0.075 (3)	0.039 (0)	0.155 (9)
8	0.139 (0)	0.070 (4)	0.041 (0)	0.155 (1)
9	0.140 (1)	0.070 (1)	0.044 (0)	0.157 (5)
MDL estimation	0.122	0.060	0.022	0.117
Comparison method	0.120	0.047	0.028	0.117

Table 4: Integral error $\mathcal{E}$ of the proposed method averaged over 100 trials to estimate a two-step change pattern (the value in parentheses is the number of trials where the obtained pattern took the minimal MDL value at K).

♯change points (K)	Blog	Wikipedia	Enron	Coauthorship
0	1.500 (0)	0.358 (1)	0.750 (0)	3.837 (0)
1	1.725 (0)	0.379 (0)	0.420 (0)	3.721 (0)
2	0.871 (0)	0.213 (18)	0.324 (0)	1.889 (0)
3 ($= K^*$)	0.133 (95)	0.138 (37)	0.128 (12)	0.279 (32)
4	0.135 (3)	0.116 (18)	0.057 (18)	0.157 (37)
5	0.135 (2)	0.113 (10)	0.052 (20)	0.149 (15)
6	0.135 (0)	0.107 (8)	0.046 (29)	0.154 (9)
7	0.135 (0)	0.107 (4)	0.047 (11)	0.155 (4)
8	0.135 (0)	0.107 (2)	0.046 (12)	0.162 (0)
9	0.135 (0)	0.107 (2)	0.047 (8)	0.169 (3)
MDL estimation	0.133	0.103	0.038	0.123
Comparison method	0.845	0.180	0.321	2.043

patterns obtained by the proposed method (shown in the row indicated by "MDL estimation") are very favorably comparable to those obtained by the comarison method that is optimized solely to a single rect-linear pattern used here. Further, the comparison method explicitly uses the constraint $p_0 = p_2$, but the proposed method does not use this constraint and estimates p_2 independenly of p_0. This implies that the pattern obtained by the proposed method can be a good approximation of the changes of the diffusion probability underlying the observed diffusion result. The number of trials where the MDL reaches the minimum is largest either at $K = 2$ or 3, which means that the MDL criteron works well to avoid an over-fitting that could be attained by introducing many

change points. There are some differences in the performace over the networks. We observe that there are more trias that the MDL criterion gives a larger K than the correct K^* for Wikipedia and Coauthorship netowrks. This is mainly attiributed to the diffusion data we used. However, more deeper alanysis is needed to understand what causes this difference, but it is true to say that the error is always small enough for the MDL results on the average.

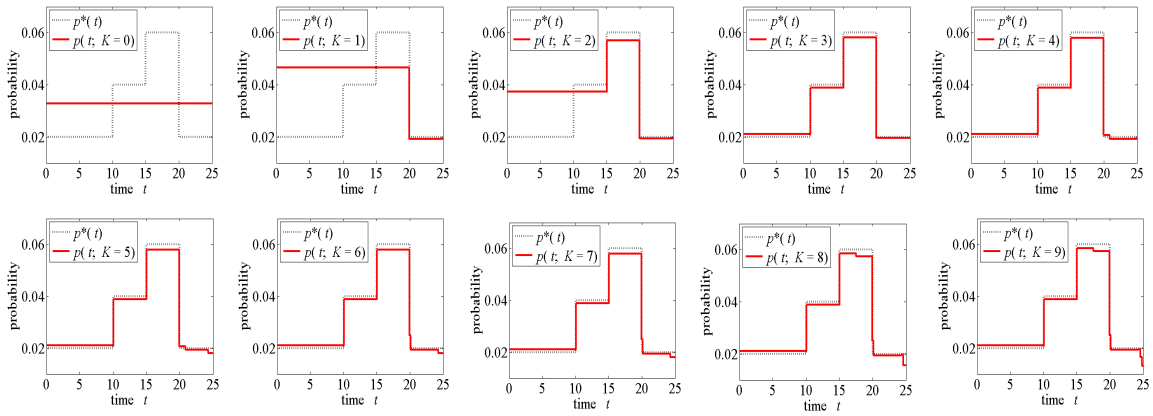
Table 4 show the results for the second (two-step) pattern. The results are qualitatively the same as in the first pattern. The estimation error drops down drastically at $K = 3(= K^*)$ and the MDL value takes the minimum at around $K = 3$ in most of the cases. For every network, the estimation errors of the optimal patterns obtained by the proposed method are about the same to those for the first pattern, and are much better than those obtained by the corresponding comparison method. In fact, it is unfair to compare the results with the comparison method because the latter is not designed to detect patterns other than the rect-linear shape, It simply shows that the comparison method cannot approximate the correct pattern by any means. The proposed method can estimate the underlying diffusion probability change in good accuracy, and the MDL based criterion to select an optimal K works well as intended also for the case of two-step pattern.

In order to analyze our experimental results more closely, we examined the diffusion probability patterns obtained by our proposed method. Figure 1 shows typical examples of desirable and undesirable cases for Wikipedia network by which a relatively large number of undesirable ones were observed. Here we simply denoted our obtained result $\hat{p}(t; \hat{\boldsymbol{t}}_k, \hat{\boldsymbol{\theta}}_k)$ as $p(t; K = k)$ for a notational convenience. From this figure, we observe that for both cases, similar change points were detected until $K \leq 2$, but their results are drastically different in the optimal number of change points, $K = K^* = 3$. In the desirable case, an almost accurate change point around $t = 10$ was detected at $K = 3$, and after that, several change points that bring about over-fitting results were detected. Actually, in terms of the MDL criterion, we could obtain the optimal number of change points and a reasonably accurate diffusion probability pattern in this case. On the other hand, in the undesirable case, a change point that brings about over-fitting results was detected at $K = 3$. At $K = 5$, a change point between $t = 10$ and 15 was detected, but this point is not so accurate compared to the point detected in the desirable case. The main reason why such undesirable cases happen for Wikipedia network is that for a relatively large number of diffusion results generated by using this network, the numbers of active nodes at an early period before $t = 10$ was quite small due to our setting of the diffusion probability $p_0 = 0.02$, which is small. As for the comparison method shown in case of the rect-linear shape in Table 3, we consider that this problem caused by small numbers of active nodes at an early period was alleviated by the imposed constraint $p_0 = p_2$.

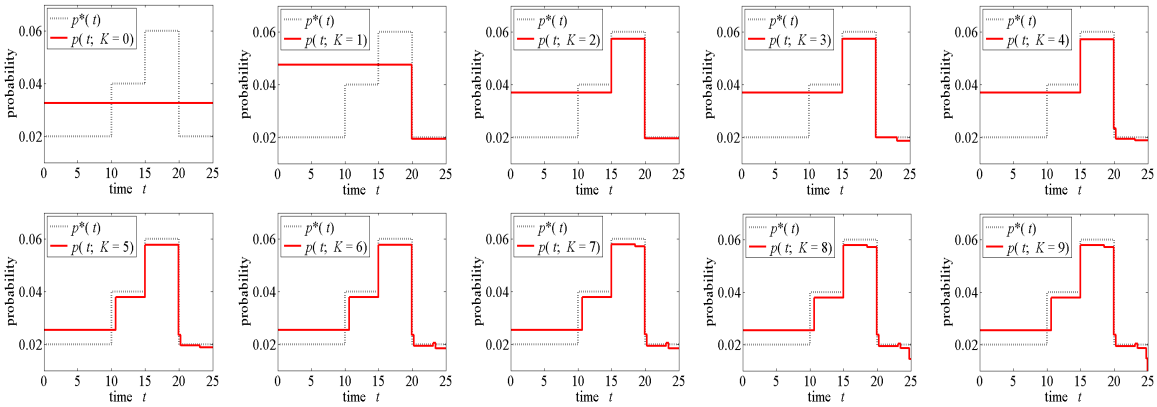
In summary, we can say that the proposed method can approximate the changes of diffusion probability underlying the observed diffusion result in good accuracy, and the MDL criteron helps avoid the over-fitting.

7. Conclusion

We addressed the problem of estimating diffusion probability changes, which are caused by changes in unknown external factors, for the AsIC-SIS (Asynchronous Indepdendent Cascade - Susceptible/Infectious/Susceptible) model over a social network from an observed information diffusion sequence. Here, the AsIC-SIS model is an information diffusion model in which the well-known discrete time IC-SIR (Independent Cascade - Susceptible/Infectious/Recovered) model is extended to contiuous time model allowing asynchronous time-delay as well as allowing multiple activations



(a) Desirable Case for Wikipedia Network



(b) Undesirable Case for Wikipedia Network

Figure 1: Examples of Diffusion Probability Functions Obtained by Varying K.

of the same nodes. We assumed that the change pattern of diffusion parameter for the AsIC-SIS model is approximated by a series of step functions, and proposed a method for detecting how many step functions are needed, where in time each one starts and how long it lasts, and what the height of each one is, from an observed sequence of information diffusion under the AsIC-SIS model. The proposed method employs "model parameter learning" by maximizing the likelihood function of the observed data (which is embedded inside the pattern search loop) and "efficient search" that uses the first order derivative of the likelihood function with respect to the parameters as a primary guide to search. The search algorithm adopts a divide-and-conquer type greedy recursive partioning that requires the expensive parameter learning only once for each partitioning, and utilizes an MDL selection measure to determine the adequate number of step functions, *i.e.*, when to stop the search. Using four real world network structures, we confirmed the effectiveness of the proposed method. We evaluated the performance of the proposed method in terms of the L^1 norm of the difference between the true and the estimated diffusion probability patterns. We tested two kinds of artificially generated change pattern: One is a rect-linear pattern having two change points, and the other is a two-step pattern having three change points. For the rect-linear pattern, the perfomance of the proposed method was very close to that of the existing method which was devised soley for this

restricted change pattern and known to work well. The performance of the proposed method for the two-step pattern did not degrade and the errors were comparable to those for the rect-linear pattern. The MDL criterion was useful to decide when to stop the search in order to avoid overfitting, and it identified the correct number of step functions in many cases. It returned a slightly large number in some cases, but the the L^1 norm of the difference between the two patterns which we use as a measure for the goodness of the found pattern was always small. Since the diffusion probability may change abruptly or gradually over time, our immediate future work is to evaluate the proposed method for a wide range of change patterns over time and reenforce the results obtained in this paper. Another immediate future work is to do a deeper analysis about why different networks give different results and understand the key factors to explain this.

Appendix A. Estimation Algorithm for AsIC-SIS Model

We briefly describe the estimation algorithm of parameters p and r for the AsIC-SIS model from an observed data $\mathcal{D}(0,T)$ (see Saito et al. (2009b, 2010a) for more details about the parameter learning algorithm of the AsIC model).

We employ an EM-like algorithm. Let $\bar{p}$ and $\bar{r}$ be the current estiamtes of p and r. Using Eqs. (2) and (3), we define $\bar{\alpha}_{u(\zeta),v(\eta)}$ and $\bar{\beta}_{u(\zeta),v(\eta)}$ as follows:

$$\begin{aligned}\alpha_{u(\zeta),v(\eta)} &= \frac{\mathcal{X}_{u(\zeta),v(\eta)}(\bar{p},\bar{r})/\mathcal{Y}_{u(\zeta),v(\eta)}(\bar{p},\bar{r})}{\sum_{z(\xi)\in\mathcal{AP}_{v(\eta)}}\mathcal{X}_{z(\xi),v(\eta)}(\bar{p},\bar{r})/\mathcal{Y}_{z(\xi),v(\eta)}(\bar{p},\bar{r})}\\ \beta_{u(\zeta),v(\eta)} &= \frac{\bar{p}\exp(-\bar{r}(t_{v(\eta)}-t_{u(\zeta)}))}{\mathcal{Y}_{u(\zeta),v(\eta)}(\bar{p},\bar{r})}\end{aligned}$$

The update formulas of p and r are as follows:

$$\begin{aligned}p &= \frac{\sum_{v(\eta)\in\mathcal{D}}\sum_{u(\zeta)\in\mathcal{AP}_{v(\eta)}}\left(\bar{\alpha}_{u(\zeta),v(\eta)}+(1-\bar{\alpha}_{u(\zeta),v(\eta)})\bar{\beta}_{u(\zeta),v(\eta)}\right)}{\sum_{u(\zeta)\in\mathcal{D}}|F(u(\zeta))|}\\ r &= \frac{\sum_{v(\eta)\in\mathcal{D}}\sum_{u(\zeta)\in\mathcal{AP}_{v(\eta)}}\bar{\alpha}_{u(\zeta),v(\eta)}}{\sum_{v(\eta)\in\mathcal{D}}\sum_{u(\zeta)\in\mathcal{AP}_{v(\eta)}}\left(\bar{\alpha}_{u(\zeta),v(\eta)}+(1-\bar{\alpha}_{u(\zeta),v(\eta)})\bar{\beta}_{u(\zeta),v(\eta)}\right)(t_{v(\eta)}-t_{u(\zeta)})}.\end{aligned}$$

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 23500312).

References

D. Crandall, D. Cosley, D. Huttenlocner, J. Kleinberg, and S. Suri. Feedback effects between similarity and sociai infiuence in online communities. In *Proceedings of KDD 2008*, pages 160–168, 2008.

P. Domingos. Mining social networks for viral marketing. *IEEE Intelligent Systems*, 20:80–82, 2005.

J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12:211–223, 2001.

M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010)*, pages 1019–1028, 2010.

D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. *SIGKDD Explorations*, 6:43–52, 2004.

D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, pages 137–146, 2003.

M. Kimura, K. Saito, and H. Motoda. Minimizing the spread of contamination by blocking links in a network. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08)*, pages 1175–1180, 2008.

M. Kimura, K. Saito, and H. Motoda. Blocking links to minimize contamination spread in a social network. *ACM Transactions on Knowledge Discovery from Data*, 3:9:1–9:23, 2009a.

M. Kimura, K. Saito, and H. Motoda. Efficient estimation of influence functions fot sis model on social networks. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, 2009b.

M. Kimura, K. Saito, R. Nakano, and H. Motoda. Extracting influential nodes on a social network for information diffusion. *Data Min. Knowl. Disc.*, 20:70–97, 2010a.

M. Kimura, K. Saito, K. Ohara, and H. Motoda. Learning to predict opinion share in social networks. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-10)*, pages 1364–1370, 2010b.

B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *Proceedings of the 2004 European Conference on Machine Learning (ECML'04)*, pages 217–226, 2004.

J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06)*, pages 228–237, 2006.

S. A. Myers and J. Leskovec. On the convexity of latent social network inference. In *Proceedings of Neural Information Processing Systems (NIPS)*.

M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

M. E. J. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 66:035101, 2002.

K. Ohara, K. Saito, M. Kimura, and H. Motoda. Efficient detection of hot span in information diffusion from observation (to appear). In *Proceedings of the IJCAI Workshop on Link Analysis in Heterogeneous Information Networks (HINA2011)*, 2011.

G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

K. Saito, M. Kimura, R. Nakano, and H. Motoda. Finding influential nodes in a social network from information diffusion data. In *Proceedings of the International Workshop on Social Computing and Behavioral Modeling (SBP09)*, pages 138–145, 2009a.

K. Saito, M. Kimura, K. Ohara, and H. Motoda. Learning continuous-time information diffusion model for social behavioral data analysis. In *Proceedings of the 1st Asian Conference on Machine Learning (ACML2009)*, pages 322–337. LNAI 5828, 2009b.

K. Saito, M. Kimura, K. Ohara, and H. Motoda. Behavioral analyses of information diffusion models by observed data of social network. In *Proceedings of the 2010 International Conference on Social Computing and Behavioral Modeling (SBP10)*, pages 149–158, 2010a.

K. Saito, M. Kimura, K. Ohara, and H. Motoda. Selecting information diffusion models over social networks for behavioral analysis. In *Proceedings of the 2010 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2010)*, pages 180–195. LNAI 6323, 2010b.

D. J. Watts. A simple model of global cascades on random networks. *Proceedings of National Academy of Science, USA*, 99:5766–5771, 2002.

D. J. Watts and P. S. Dodds. Influence, networks, and public opinion formation. *Journal of Consumer Research*, 34:441–458, 2007.

F. Wu and B. A. Huberman. How public opinion forms. In *Proceedings of WINE 2008*, pages 334–341, 2008.

Learning Attribute-weighted Voter Model over Social Networks

Yuki Yamagishi B08107@U-SHIZUOKA-KEN.AC.JP
School of Management and Information
University of Shizuoka

Kazumi Saito K-SAITO@U-SHIZUOKA-KEN.AC.JP
School of Management and Information
University of Shizuoka

Kouzou Ohara OHARA@IT.AOYAMA.AC.JP
Department of Integrated Information Technology
Aoyama Gakuin University

Masahiro Kimura KIMURA@RINS.RYUKOKU.AC.JP
Department of Electronics and Informatics
Ryukoku University

Hiroshi Motoda MOTODA@AR.SANKEN.OSAKA-U.AC.JP
Institute of Scientific and Industrial Research
Osaka University

Editor:

Abstract

We propose an opinion formation model, an extension of the voter model that incorporates the strength of each node, which is modeled as a function of the node attributes. Then, we address the problem of estimating parameter values for these attributes that appear in the function from the observed opinion formation data and solve this by maximizing the likelihood using an iterative parameter value updating algorithm, which is efficient and is guaranteed to converge. We show that the proposed algorithm can correctly learn the dependency in our experiments on four real world networks for which we used the assumed attribute dependency. We further show that the influence degree of each node based on the extended voter model is substantially different from that obtained assuming a uniform strength (a naive model for which the influence degree is known to be proportional to the node degree), and is more sensitive to the node strength than the node degree even for a moderate value of the node strength.

Keywords: voter model, influence degree, attribute dependency.

1. Introduction

The growth of Internet has enabled to form various kinds of large-scale social networks, and a variety of information, e.g. news, ideas, hot topics, rumors, etc. spreads in the form of "word-of-mouth" communications. It is noticeable to observe how much they affect our daily life style. The spread of information has been studied by many researchers (Newman et al., 2002; Newman, 2003; Gruhl et al., 2004; Domingos, 2005; Leskovec et al., 2006; Kimura et al., 2009, 2010a). The information diffusion models widely used are the *independent cascade (IC)* (Goldenberg et al., 2001; Kempe et al., 2003; Kimura et al., 2009) and the *linear threshold (LT)* (Watts, 2002; Watts and Dodds, 2007)

models. Both have been used to solve such problems as the *influence maximization problem* (Kempe et al., 2003; Kimura et al., 2007) and the *contamination minimization problem* (Kimura et al., 2009; Tong et al., 2010). These two models focus on different information diffusion aspects. The IC model is sender-centered and each active node *independently* influences its inactive neighbors with given diffusion probabilities. The LT model is receiver-centered and a node is influenced by its active neighbors if their total weight exceeds the threshold for the node. Thus, it can be said that the IC model emphasizes "information push" and the LT model "information pull".

In this paper, we address a different kind of information diffusion, which is "opinion formation", *i.e.*, spread of opinions. A well studied model for opinion dynamics is the voter model which has the same key property with the LT model that a node decision is influenced by its neighbor's decision, *i.e.*, a person changes his or her opinion by the opinions of his or her neighbors. The basic voter model is defined on an undirected network and allows to have only two opinions. Each node adopts the opinion of a randomly chosen neighbor at each subsequent discrete time-step. There has been a variety of work on the voter model. Dynamical properties of the basic model, including how the degree distribution and the network size affect the mean time to reach consensus, have been extensively studied (Liggett, 1999; Sood and Redner, 2005) from mathematical point of view. Several variants of the voter model are also investigated (Castellano et al., 2009; Yang et al., 2009) and non equilibrium phase transition is analyzed from physics point of view. Yet another line of work extends the voter model and combines it with a network evolution model (Holme and Newman, 2006; Crandall et al., 2008).

The work which is most influential to this work is by Even-Dar and Shapira (Even-Dar and Shapria, 2007) who investigated the influence maximization problem at a given target time. They showed that the most natural heuristic solution, which picks the nodes in the network with the highest degree, is indeed the optimal solution. We extended the basic voter model to be able to handle multiple opinions and asynchronous time delay and, in doing so, introduced the value for each opinion to reflect the fact that people are affected by the importance of the opinion, *e.g.*, quality, brand, authority, etc. (Kimura et al., 2010b). We called this model "Value-weighted Voter Model with Multiple Opinions (VwVM)", and addressed the problem of predicting the expected opinion share at a target time from the observed opinion formation data. We further addressed the problem of detecting the change of the opinion values from the observed data (Saito et al., 2011)

In this work, we introduce another factor which we call *strength* of each node. This is different from the *value* of opinion that was introduced in Kimura et al. (2010b). It is based on the observation that a person is influenced not only by what each opinion is about but also by who holds/says that opinion. Some persons are more influential than others, and we consider this degree of influence by the strength value that is associated with each node. Here, we must note that the *influence* meant here can better be named as direct influence and is different from what has been used in previous studies which can better be named as indirect influence. In information diffusion the *influence degree* of a node is defined as the expected number of active nodes at the end of the random process of the information diffusion that originated from the node (Kempe et al., 2003; Kimura et al., 2007). In particular, in opinion formation, it is defined as the expected number of nodes that hold the same opinion with the starting node at the end of the random process of the opinion formation (Even-Dar and Shapria, 2007). We distinguish the strength of a node, *i.e.*, direct influence, from the influence degree of the node, *i.e.*, indirect influence. The strength we define here is assumed to be intrinsic to each node and is determined independently of the result of information diffusion or opinion formation.

The problem we want to solve in this paper is to learn this strength from the observed opinion formation data and investigate how it affects the influence degree. In principle it is possible to learn the strength of all the nodes in the network from the observed data, given the generative model of opinion formation, by maximizing the likelihood of the observed data being generated. However, the number of nodes is huge and we need prohibitively large amount of observation data to avoid the overfitting problem. We rather assume and think it more natural that the strength is determined by the attributes of each node and its attribute dependency is more or less uniform across the nodes, and try to learn the parameters that define this attribute dependency from the data. We call this model " Attribute-weighted Voter Model with Multiple Opinions (AwVM)" in contrast to the model we previously defined, "Value-weighted Voter Model with Multiple Opinions (VwVM)".

We derived a very efficient parameter updating algorithm to maximize the likelihood function that is guaranteed to converge, and tested the performance of the algorithm on four real world networks assuming the attribute dependency of the parameters to be of a particular form. The algorithm can correctly estimate the strength of each node by way of node attributes through a learned function. We further show that the influence degree of each node based on the AwVM is substantially different from a naive AwVM that assumes a uniform strength throughout the nodes for which the influence degree is known to be proportional to the node degree, and there appears to be no simple heuristic to approximate the influence degree with good accuracy unless the network is dense. The sensitivity analysis indicates that as the degree of non-uniformity of the strength becomes greater, the influence degree becomes progressively more sensitive to the node strength than the node degree, and even for a moderate value of the non-uniformity of node strength, it is more affected by the node strength than by the node degree.

2. Opinion Formation Models

Let $G = (V, E)$ be an undirected (bidirectional) network with self-loops, where V and E ($\subset V \times V$) are the sets of all the nodes and links in the network, respectively. For a node $v \in V$, let $\Gamma(v)$ denote the set of neighbors of v in G, that is, $\Gamma(v) = \{u \in V; (u, v) \in E\}$. Note that $v \in \Gamma(v)$.

2.1 Basic Voter Model

According to the work of Even-Dar and Shapria (2007), we recall the definition of the basic voter model with two opinions on networks G. In the voter model, each node of G is endowed with two states; opinions 1 and 2. The opinions are initially assigned to all the nodes in G, and the evolution process unfolds in discrete time-steps $t = 1, 2, 3, \cdots$ as follows: At each time-step t, each node v picks a random neighbor u and adopts the opinion that u holds at time-step $t - 1$.

More formally, let $f_t : V \to \{1, 2\}$ denote the opinion distribution at time-step t, where $f_t(v)$ stands for the opinion of node v at time-step t. Then, $f_0 : V \to \{1, 2\}$ is the initial opinion distribution, and $f_t : V \to \{1, 2\}$ is inductively defined as follows: For any $v \in V$,

$$\begin{cases} f_t(v) = 1, & \text{with probability } |n_1(t, v)|/|\Gamma(v)|, \\ f_t(v) = 2, & \text{with probability } |n_2(t, v)|/|\Gamma(v)|, \end{cases}$$

where $n_k(t, v) = \{u \in \Gamma(v); f_{t-1}(u) = k\}$ is the set of v's neighbors that hold opinion k at time-step $t - 1$ for $k = 1, 2$.

Again, according to the work of Even-Dar and Shapria (2007), we define the expected influence degree of each node v, denoted by $\sigma_b(v)$. Consider an initial opinion distribution that $f_0(v) = 1$ and $f_0(w) = 2$ if $w \neq v$. Namely, only v's opinion is 1 and that of all the other nodes is 2. Then $\sigma_b(v)$

is defined as the expected number of nodes that hold the opinion 1 (node v's initial opinion) after enough time has passed. More formally, for a given network $G = (V, E)$, we identify each node with a unique integer from 1 to $|V|$. Then we can define the adjacency matrix $\boldsymbol{A} \in \{0,1\}^{|V|\times|V|}$ by setting $a_{u,v} = 1$ if $(u, v) \in E$; otherwise $a_{u,v} = 0$. We also define the diagonal matrix $\boldsymbol{D}$, each element of which, $d_{v,v} = d(v)$, is the degree of node v, *i.e.*, $d(v) = |\Gamma(v)| = \sum_{u\in V} a_{u,v}$. Here note that $d(v) \geq 1$ due to self-loops. Let $\boldsymbol{h}_v \in \{0,1\}^{|V|}$ be a vector whose v-th element is 1 and all the other elements are 0, and $\boldsymbol{h} \in \{1\}^{|V|}$ be a vector whose elements are all 1. Now, we can calculate the expected number of nodes that hold opinion 1 at time $t = 1$ starting from a node v by $\boldsymbol{h}_v^T \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{h}$. Thus, the vector $\boldsymbol{b}$ of the expected influence degree, each element of which is $b_v = \sigma_b(v)$, is defined as a limiting solution of the following iterative process with the initial setting $\boldsymbol{b}^{(0)} = \boldsymbol{h}$.

$$\boldsymbol{b}^{(t)} = \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{b}^{(t-1)}. \tag{1}$$

Especially, in case of an undirected network, the analytical solution can be derived, *i.e.*, $b_v = \sigma_b(v) = |V||\Gamma(v)|/\sum_{u\in V}|\Gamma(u)|$. The result clearly states that the influence degree b_v of a node v is proportional to its node degree $|\Gamma(v)|$.

2.2 Attribute-weighted Voter Model

We extend the basic voter model by allowing to hold K opinions ($K \geq 2$). Further as explained in Section 1, we introduce the strength for each node v, denoted by s_v[1]. Then, we can define the following probability of opinion adoption for the new voter model with the node strength.

$$P(f_t(v) = k) = \frac{\sum_{u\in n_k(t,v)} s_u}{\sum_{u\in\Gamma(v)} s_u}, \quad (k = 1, \cdots, K), \tag{2}$$

where $n_k(t, v) = \{u \in \Gamma(v); f_{t-1}(u) = k\}$ is the set of v's neighbors that hold opinion k at time-step $t-1$ for $k = 1, 2, \cdots, K$.

Similarly to the basic voter model, we can define the expected influence degree $\sigma_a(v)$ for the new voter model, which is the expected number of nodes that hold the opinion k after enough time has passed when only the node v has the opinion k and all the other nodes have different opinions at $t = 0$. To this end, we define the diagonal matrix $\boldsymbol{W}$, each diagonal element of which, $w_{v,v} = w(v)$, represents the total strength of v's neighbors, $w(v) = \sum_{u\in\Gamma(v)} s_u$. According to the arguments of the basic voter model, the vector $\boldsymbol{a}$ of the expected influence degree, each element of which is $a_v = \sigma_a(v)$, is defined as a limiting solution of the following iterative process with the initial setting $\boldsymbol{a}^{(0)} = \boldsymbol{h}$.

$$\boldsymbol{a}^{(t)} = \boldsymbol{S}\boldsymbol{W}^{-1}\boldsymbol{a}^{(t-1)}, \tag{3}$$

where $\boldsymbol{S}$ is the strength matrix which is obtained by replacing $a_{u,v}$ of $\boldsymbol{A}$ with $a_{u,v}s_u$. Note that unlike Eq. (1), no analytical solution is known for Eq. (3), and how the strength s_v affects the influence degree a_v is not clear. Thus, we solve it numerically by iteratively calculating Eq. (3) until the difference $\|\boldsymbol{a}^{(t)} - \boldsymbol{a}^{(t-1)}\|$ becomes a reasonably small value. The convergence is guaranteed by the Perron-Frobenius theorem because the network is connected. Further note that Eq. (3) is defined independently of the number of opinions K.

In general, we don't know the adequate value for the strength of each node apriori. As one possible approach, we consider estimating each of them from a set of observed opinion formation

1. One of the anonymouse reviewers pointed that our formulation is similar to the policy gradient learning in the reinforcement learning, which we were not aware of, *i.e.*, Eq. (2) corresponds to a Boltzman distribution policy, Eq. (3) to the transition model of a Markov decision problem (MDP), and estimating the strength of a node to learning the value function of the MDP.

results. However, as each node has its own strength, the number of parameters to learn is so huge that we need prohibitively large amount of training data to learn them all individually. As described in Section 1, we note that it is more natural to think that the strength is determined by the attributes of each node (person), *e.g.*, occupation, physical appearance, income, social status, etc., and its attribute dependency is more or less uniform across the nodes. We, thus, propose an Attribute-weighted Voter Model with Multiple Opinions (AwVM) that explicitly considers the dependency of node strength on its attributes. We assume that each node can have multiple attributes. Let $x_{v,j}$ be a value that node v takes for the j-th attribute, and J the total number of the attributes. We denote the J-dimensional vector of the attribute values for each node v by $\boldsymbol{x}_v$. Then we propose to model the strength s_v of node v by the following formula [2]:

$$s_v \;=\; s(\boldsymbol{x}_v, \boldsymbol{\theta}) \;=\; \exp(\boldsymbol{\theta}^T \boldsymbol{x}_v), \tag{4}$$

where $\boldsymbol{\theta}^T = (\theta_1, \cdots, \theta_J)$ is the J-dimensional parameter vector for the attributes to determine the strength value of each node.

So far, we assumed a discrete time step. However, the actual opinion formation takes place in an asynchronous way along the continuous time axis, and the time stamps of the observed data are not equally spaced. Thus, there is a need to extend the model to make the state changes asynchronous. In order to describe the asynchronous voter model, we need to extend the definition of $n_k(t, v)$, the set of v's neighbors that hold opinion k at $t-1$, to be the one with the latest opinion before time t, *i.e.*,

$$n_k(t, v) = \{u \in \Gamma(v);\ \varphi_t(u) = k\},$$

where $\varphi_t(u)$ is the latest opinion of u before time t.[3] Then, the evolution process of the asynchronous voter model is defined as follows:

1. At time 0, each node v independently decides its update time t according to some probability distribution such as an exponential distribution with parameter 1.[4] The successive update time is determined similarly at each update time t.
2. At update time t, the node v adopts a new opinion according to Eq. (2).
3. The process is repeated from the initial time $t = 0$ until the next update-time exceeds a given final-time T.

3. Learning Problem and Method

We consider the problem of estimating the parameters for the attributes from observed data $\mathcal{D}_T$ in time-span $[0, T]$, where $\mathcal{D}_T$ consists of a sequence of (v, t, k) such that node v updated its opinion to opinion k at time t for $0 \le t \le T$.[5] By estimating parameters, we can identify the influential nodes, *i.e.*, those with large influence degree, as well as the relevant attributes for determining the strength of nodes.

We formulate our problem for estimating the parameter values of the AwVM from a given observed opinion formation data $\mathcal{D}_T$. Based on the evolution process of our model (see Eq. (2)), we

2. This is a simple smooth function with respect to $\boldsymbol{\theta}$ that guarantees $s_v > 0$.
3. Note that the opinion update takes place at time t and we need the distribution before the time t.
4. This assumes that the average delay time is 1.
5. The data come in sequence each time the update takes place, but in the formulation we treat them as a set for easiness of the mathematical treatment.

can obtain the log likelihood function,

$$\begin{aligned}\mathcal{L}(\mathcal{D}_T;\boldsymbol{\theta}) &= \log\left(\prod_{(v,t,k)\in\mathcal{D}_T} P(f_t(v)=k)\right)\\ &= \sum_{(v,t,k)\in\mathcal{D}_T}\left(\log\left(\sum_{u\in n_k(t,v)}\exp(\boldsymbol{\theta}^T\boldsymbol{x}_u)\right)-\log\left(\sum_{u\in\Gamma(v)}\exp(\boldsymbol{\theta}^T\boldsymbol{x}_u)\right)\right).\end{aligned} \tag{5}$$

Thus our estimation problem is formulated as a maximization problem of the objective function $\mathcal{L}(\mathcal{D}_T;\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

We derive an EM like iterative algorithm for obtaining the maximum likelihood estimators. Now, let $\bar{\boldsymbol{\theta}}$ be the current estimates of $\boldsymbol{\theta}$. Then, by considering the posterior probabilities,

$$q_{v,t,k,u}(\boldsymbol{\theta}) = \frac{\exp(\boldsymbol{\theta}^T\boldsymbol{x}_u)}{\sum_{w\in n_k(t,v)}\exp(\boldsymbol{\theta}^T\boldsymbol{x}_w)},$$

($v\in V$, $0\le t\le T$, $k=1,\cdots,K$, $u\in n_k(t,v)$), we can transform our objective function as follows:

$$\mathcal{L}(\mathcal{D}_T;\boldsymbol{\theta}) = Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}}) - \mathcal{H}(\boldsymbol{\theta};\bar{\boldsymbol{\theta}}), \tag{6}$$

where $Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})$ is defined by

$$Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}}) = \sum_{(v,t,k)\in\mathcal{D}_T}\left(\left(\sum_{u\in n_k(t,v)} q_{v,t,k,u}(\bar{\boldsymbol{\theta}})\,\boldsymbol{\theta}^T\boldsymbol{x}_u\right)-\log\left(\sum_{u\in\Gamma(v)}\exp(\boldsymbol{\theta}^T\boldsymbol{x}_u)\right)\right), \tag{7}$$

and $\mathcal{H}(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})$ is defined by

$$\mathcal{H}(\boldsymbol{\theta};\bar{\boldsymbol{\theta}}) = \sum_{(v,t,k)\in\mathcal{D}_T}\sum_{u\in n_k(t,v)} q_{v,t,k,u}(\bar{\boldsymbol{\theta}})\log q_{v,t,k,u}(\boldsymbol{\theta}). \tag{8}$$

It is self-evident that $\mathcal{H}(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})$ is maximized when $\boldsymbol{\theta}=\bar{\boldsymbol{\theta}}$, which is easily proved by noting the constraint $\sum_u q_{v,t,k,u}(\bar{\boldsymbol{\theta}}) = 1$. Then, it follows $\mathcal{L}(\mathcal{D}_T;\boldsymbol{\theta}) - \mathcal{L}(\mathcal{D}_T;\bar{\boldsymbol{\theta}}) \ge Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}}) - Q(\bar{\boldsymbol{\theta}};\bar{\boldsymbol{\theta}})$. Thus, $\mathcal{L}(\mathcal{D}_T;\boldsymbol{\theta})$ always increases by repeatedly maximizing $Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})$ with respect to $\boldsymbol{\theta}$ and updating $\bar{\boldsymbol{\theta}}$. When $Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})$ reaches a fixed point, *i.e.*, Q is the maximum at $\boldsymbol{\theta}=\bar{\boldsymbol{\theta}}$ for $\bar{\boldsymbol{\theta}}$, it holds that $(\partial Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})/\partial\boldsymbol{\theta})_{\boldsymbol{\theta}=\bar{\boldsymbol{\theta}}} = (\partial\mathcal{L}(\mathcal{D}_T;\boldsymbol{\theta})/\partial\boldsymbol{\theta})_{\boldsymbol{\theta}=\bar{\boldsymbol{\theta}}} = 0$, *i.e.*, when Q reaches a fixed point, $\mathcal{L}(\mathcal{D}_T;\boldsymbol{\theta})$ also reaches a maximum, not neccessarily a global maximum.

In order to derive our maximization algorithm for $Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})$, we further define the following probability that the node v adopts the opinion of node u at time t whatever opinion u has:

$$r_{v,t,u}(\boldsymbol{\theta}) = \frac{\exp(\boldsymbol{\theta}^T\boldsymbol{x}_u)}{\sum_{w\in\Gamma(v)}\exp(\boldsymbol{\theta}^T\boldsymbol{x}_w)}.$$

Then, we can obtain the gradient vector of $Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})$ with respect to $\boldsymbol{\theta}$ as follows:

$$\frac{\partial Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}} = \sum_{(v,t,k)\in\mathcal{D}_T}\left(\left(\sum_{u\in n_k(t,v)} q_{v,t,k,u}(\bar{\boldsymbol{\theta}})\,\boldsymbol{x}_u\right)-\left(\sum_{u\in\Gamma(v)} r_{v,t,u}(\boldsymbol{\theta})\,\boldsymbol{x}_u\right)\right).$$

Similarly, we can obtain the Hessian matrix with respect to $\boldsymbol{\theta}$ as follows:

$$\frac{\partial^2 Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T} = -\sum_{(v,t,k)\in\mathcal{D}_T}\left(\left(\sum_{u\in\Gamma(v)} r_{v,t,u}(\boldsymbol{\theta})\,\boldsymbol{x}_u\boldsymbol{x}_u^T\right)-\left(\sum_{u\in\Gamma(v)} r_{v,t,u}(\boldsymbol{\theta})\,\boldsymbol{x}_u\right)\left(\sum_{u\in\Gamma(v)} r_{v,t,u}(\boldsymbol{\theta})\,\boldsymbol{x}_u\right)^T\right).$$

Thus we can obtain the following modification vector $\Delta\boldsymbol{\theta}$ for updating $\boldsymbol{\theta}$:

$$\Delta\boldsymbol{\theta} = -\left(\frac{\partial^2 Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\right)^{-1}\frac{\partial Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}}. \tag{9}$$

Here note that the following quadratic form of the Hessian matrix is non-positive for an arbitrary J-dimensional non-zero vector $\boldsymbol{z} = (z_1, \cdots, z_J)$,

$$\boldsymbol{z}^T\frac{\partial^2 Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\boldsymbol{z} = -\sum_{(v,t,k)\in\mathcal{D}_T}\left(\left(\sum_{u\in\Gamma(v)} r_{v,t,u}(\boldsymbol{\theta})\,(\boldsymbol{x}_u^T\boldsymbol{z})^2\right)-\left(\sum_{u\in\Gamma(v)} r_{v,t,u}(\boldsymbol{\theta})\,\boldsymbol{x}_u^T\boldsymbol{z}\right)^2\right) \le 0$$

The Hessian matrix of Q is non-positive definite, and thus, the optimal solution of Q is obtained by using the Newton method. We can regard our estimation method as a variant of the EM algorithm. Namely, calculating Q by Eq. (7) and updating $\boldsymbol{\theta}$ by Eq. (9) correspond to Expectation and Maximization steps, respectively. Which one in the $|n_k(t,v)|$ active parents actually activated its child is not known when $|n_k(t,v)| \ge 2$, which corresponds to the existence of the latent vairables although they are not explicit in our formulation. We want to emphasize here that each time iteration proceeds the value of the likelihood function never decreases and the iterative algorithm is guaranteed to converge due to the convexity of Q.

Below we summarize the algorithm of the proposed method.

1. Initialize parameter vector $\boldsymbol{\theta}$ as $\theta_j = 0$ for $j = 1, \cdots, J$.
2. Calculate the gradient vector at the current parameter vector $\boldsymbol{\theta}$.
3. If the gradient vector is sufficiently small, *i.e.*, $\|\partial Q(\boldsymbol{\theta};\bar{\boldsymbol{\theta}})/\partial\boldsymbol{\theta}\| < \eta$, output the parameter vector $\bar{\boldsymbol{\theta}}$, and then terminate. Otherwise, go to 4.
4. Update the parameter vector $\boldsymbol{\theta}$ by Eq. (9), and return to 2.

Here η is a parameter for the termination condition. In our experiments, η is set to a sufficiently small number, *i.e.*, $\eta = 10^{-12}$.

Finally, we briefly discuss the computational complexity of the learning algorithm. Evidently the most computationally expensive part is the Hessian matrix, *i.e.*, for each $(v,t,k) \in \mathcal{D}_T$ and its corresponding neighbor $u \in \Gamma(v)$, the required computational complexity is the square of the parameter size J. The average number of neighbors $\Gamma(v)$ is $|E|/|V|$, and the expected number of elements in $\mathcal{D}_T$ is $T|V|$ (this is true for the exponential distribution with parameter 1). Thus, let M be the number of iterations required to obtain the solution; then the computational complexity of the algorithm is given by $O(MT|E|J^2)$.

4. Experiments

First, we evaluated experimentally our learning algorithm using synthetic opinion formation data that were generated from four large real world networks for which we assumed a specific relation between the node attributes and the node strength (Eq. (4)). Second, we evaluated how the node strength affects the influence degree (the expected number of nodes that hold the same opinion with the starting node at the end of the random process of the opinion formation).

4.1 Network Datasets

We used four large real networks, which are all bidirectionally connected. The first one is a reader network of "Ameba"[6] that is a Japanese blog service site. We crawled the reader lists of 117,374 blogs of the Ameba blog service site in June 2006, and collected a large connected network, which has 56,604 nodes and 1,071,080 directed links (the Ameblo network). The second one is a trackback network of Japanese blogs (Kimura et al., 2009) that has $12,047$ nodes and $79,920$ directed links (the blog network). The third one is a Coauthorship network (Palla et al., 2005) and has $12,357$ nodes and $38,896$ directed links (the coauthorship network). The last one is a network of people that was derived from the "list of people" within Japanese Wikipedia (Kimura et al., 2008), which has $9,481$ nodes and $245,044$ directed links (the Wikipedia network).

4.2 Experimental Setting

For each network we generated synthetic opinion formation data $\mathcal{D}_T$ of time span $[0, T]$ in the following way: 1) artificially generate node attributes and determine their values in a random manner; 2) determine a parameter vector $\boldsymbol{\theta}$ which is assumed to be true; and then 3) generate $\mathcal{D}_T$ multiple times for the given K and T by running the true AwVM that uses $\boldsymbol{\theta}$, each of which starts from the state in which the initial opinion of each node is selected uniformly at random from the K opinions. The values of K and T are varied as needed. We generated a total of 10 attributes for every node in each network, each with a real value of $[-1, 1]$. The true parameter vector $\boldsymbol{\theta}$ was determined so that the distribution of s_v becomes uniform, that is, the expected value of $\boldsymbol{\theta}^T \boldsymbol{x}_v$, *i.e.*, $\langle \boldsymbol{\theta}^T \boldsymbol{x}_v \rangle$ becomes 0. As one such instance, we chose the parameter vector $\boldsymbol{\theta} = (2.0, -1.0, 1.0, -2.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)$. Note that this is different from limiting the number of attributes to 4.[7]

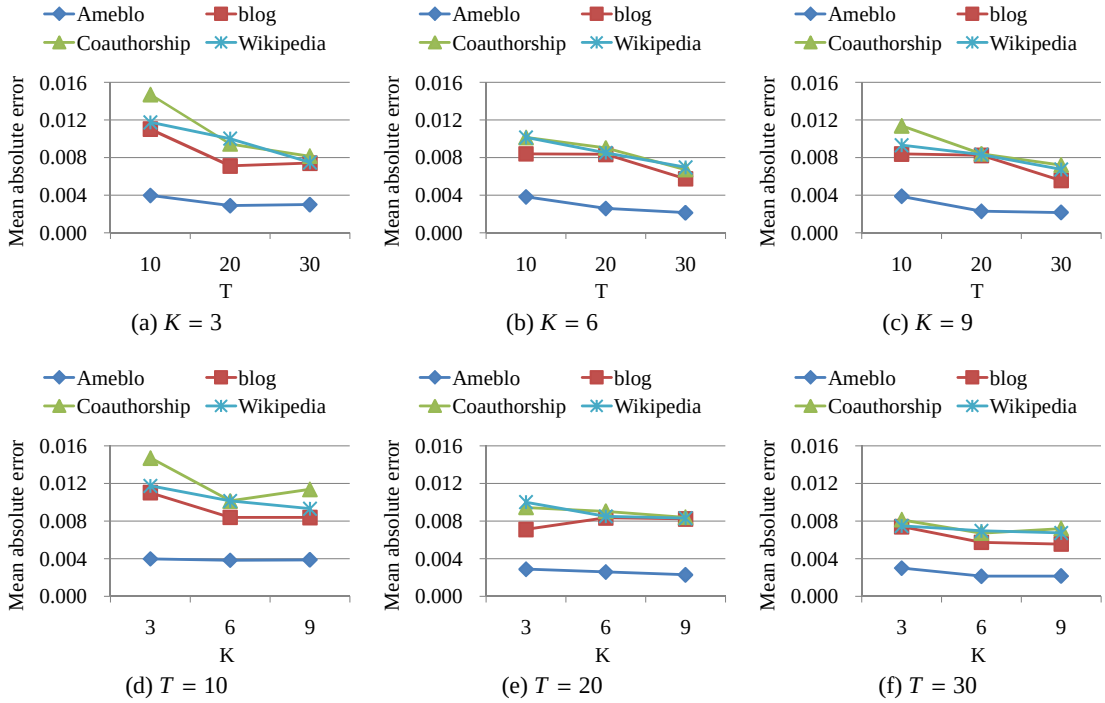
We conducted two kinds of experiments to evaluate the proposed method, one to evaluate the performance of the learning algorithm, and the other to evaluate the effect of introducing the strength on the influence degree. The learning performance was evaluated in terms of the accuracy of the parameter values $\hat{\boldsymbol{\theta}}$ that are estimated from $\mathcal{D}_T$ by our learning algorithm with different values for T and K, and its computation time. The effect of the strength was evaluated by comparing the influence degree (Eq. (3)) and the node ranking (with respect to the influence degree) of the AwVM that uses the learned parameters with the naive AwVM in which uniform strength *i.e.*, $s_v = 1$, is used, which is equivalent to the basic voter model with K opinions. Further, the node ranking of AwVM is compared with the ranking based on the node degree and other heuristic. We denote the true influence degree of AwVM as $\sigma_a(v)$ (Section 2.2), the estimated influence degree of AwVM as $\hat{\sigma}_a(v)$ and the influence degree of the naive AwVM as $\sigma_b(v)$ (Section 2.1). We terminated the iterative calculation of Eq. (3) if either of the following conditions is satisfied: the number of iteration exceeds 1,000, or $\|\boldsymbol{a}^{(t)} - \boldsymbol{a}^{(t-1)}\| < 10^{-8}$.

4.3 Evaluation of Parameter Estimation

Figure 1 shows the mean absolute error which is defined as $\Sigma_{j=1}^{J} |\theta_j - \hat{\theta}_j| / J$. It is the average over 10 trials that were conducted on the 10 distinct opinion formation data $\mathcal{D}_T$, each generated independently for each combination of the corresponding T and K for each network. Figures 1a to 1c is to show how the time span $[0, T]$ affects the results, and Figs. 1d to 1f is to show how the number of

6. http://www.ameba.jp/

7. The algorithm should be able to identify the useless attributes for which the parameter values are 0.



(a) $K = 3$ (b) $K = 6$ (c) $K = 9$

(d) $T = 10$ (e) $T = 20$ (f) $T = 30$

Figure 1: Mean absolute errors of the estimated parameter values for each network.

Table 1: Computation time (sec.) and the number of iterations of the proposed learning algorithm (in parentheses).

		$K = 3$	$K = 6$	$K = 9$
Ameblo	$T = 10$	36.49 (24.9)	22.70 (16.0)	19.49 (13.5)
	$T = 20$	71.58 (25.0)	45.70 (16.2)	40.70 (14.0)
	$T = 30$	103.51 (25.7)	70.45 (17.0)	62.85 (14.1)
blog	$T = 10$	2.24 (23.0)	1.76 (17.0)	1.58 (15.0)
	$T = 20$	4.55 (24.8)	3.61 (18.5)	3.36 (16.9)
	$T = 30$	6.77 (25.6)	5,62 (19.9)	5.26 (18.1)
Coauthorship	$T = 10$	1.06 (19.0)	0.97 (15.4)	0.90 (14.0)
	$T = 20$	2.10 (21.0)	1.91 (17.0)	1.83 (16.0)
	$T = 30$	2.97 (21.0)	2.79 (17.5)	2.71 (16.1)
Wikipedia	$T = 10$	8.07 (34.3)	5.69 (23.9)	4.95 (20.2)
	$T = 20$	17.14 (37.6)	12.75 (27.1)	11.08 (23.9)
	$T = 30$	27.84 (41.3)	19.96 (29.2)	18.55 (26.9)

opinions K affects the results. The mean absolute error is extremely small and in general decreases as both T and K increase. A larger T implies a larger training sample size of $\mathcal{D}_T$ which naturally contributes to reducing the error. A larger K implies a larger chance of updating to a different opinion at each update time which contributes to increasing the diversity of the data, again contributing to reducing the error. Said differently, when K is smaller, the time to reach local consensus gets ear-

lier and the amount of data to be effective for learning gets smaller. The error for Ameblo network being the smallest is explained by the fact that this network has by far the largest number of nodes, indicating the largest expected number of samples which is $T|V|$ as explained in the last paragraph in Section 3.

Table 1 shows the computation time. The machine used is Intel(R) Xeon(R) CPU W5590 @3.33GHz with 32GB memory. The computation time roughly follows the results of computational complexity analysis. Under the situation where the number of iteration M is about the same for the same K, which is the case here, the computation time is proportional to the number of links for the same T. Ameblo network takes 4 times longer than Wikipedia network, Wikipedia network takes 3 times longer than Blog network and Blog network takes twice longer than Coauthorship network. The number of opinions K affects favorably the computation time. This is because the larger diversity of the data as explained above accelerates the convergence of iterative procedure.

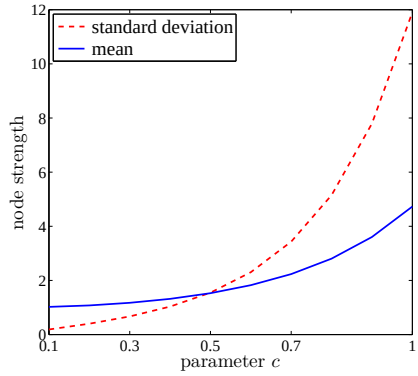
Overall, we can say that our learning method can accurately estimate the parameter values of the AwVM, and its performance with respect to the time span and the number of opinions are interpretable.

4.4 Comparative Study on Influence Degree

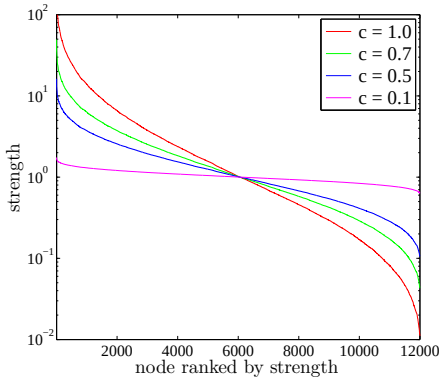
We evaluated the effect of the node strength in terms of the influence degree. To this end, we introduced a new parameter c to control the true parameter by $\boldsymbol{\theta}_c = c\boldsymbol{\theta}$. The parameter c is a measure to indicate the non-uniformity of the node strength, and is called the non-uniformity parameter. It affects the value of the strength directly, too, but as Eq.(2) shows, what matters is the relative strength. We use $c = 0.1$, 0.5, and 1.0 in the following experiments. Figure 2a illustrates how the mean of the node strength and the standard deviation (non-uniformity) change with c. Both increase exponentially as c increases. The distribution of the node strength for $c = 1.0$ is much more non-uniform than that for $c = 0.5$ as shown in Fig. 2b. The strength is almost uniform across the nodes for $c = 0.1$, but some nodes have the strength which is 100 times as high as the average for $c = 1.0$. This can be expected from the form of Eq.(4). We think such a deviation for $c = 1.0$ is not rare even in our real human relations.

Now for each c, we define the mean absolute error of the AwVM by $\epsilon_a = \frac{1}{|V|}\sum_{v\in V}|\sigma_a(v) - \hat{\sigma}_a(v)|$, and that of the naive model by $\epsilon_b = \frac{1}{|V|}\sum_{v\in V}|\sigma_a(v) - \sigma_b(v)|$. Figure 3 shows the results of the mean absolute errors ϵ_a and ϵ_b. Here, we used the 10 distinct opinion formation data, each generated with $T = 30$ and $K = 3$ for each network, and $\hat{\boldsymbol{\theta}}$ was estimated by our learning method for each trial. All of the results shown in Fig. 3 are the average over the 10 trials. The mean absolute error of the AwVM is reduced to 0.5 to 2% of the naive model for $c = 1.0$, and 10 to 20% even for $c = 0.1$ where the node strength has only a small effect (Fig. 3a). Fig. 3b shows the results of the standard deviation of the mean absolute errors. Both the mean and the standard deviation are of the same order, but the standard deviation is more sensitive to c, which can be expected by the increase in the standard deviation (non-uniformity) shown in Fig. 2a.

To do a more detailed analysis, we illustrate in Fig. 4 the actual influence degree of each node in the blog network for one particular run, randomly chosen from the 10 independent trials. The nodes in the horizontal axis are ranked in descending order of the true influence degree σ_a. Also from this result, we can find that the difference between σ_a (black solid line) and $\hat{\sigma}_a$ (red cross "×") is quite small for any value of c, while the difference between σ_a and σ_b (blue plus "+") becomes

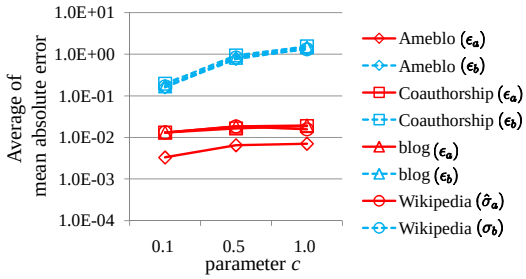


(a) Change of the mean and the standard deviation of node strength with the non-uniformity parameter c

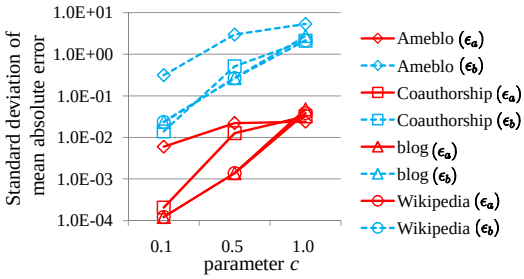


(b) The distribution of node strength with the non-uniformity parameter $c = 0.1, 0.5, 0.7, 1.0$.

Figure 2: The relation between the node strength and the non-uniformity parameter c.



(a) Average of the mean absolute errors of $\hat{\sigma}_a(v)$ and $\sigma_b(v)$



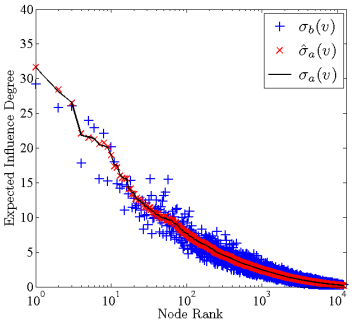
(b) Standard deviation of the mean absolute errors of $\hat{\sigma}_a(v)$ and $\sigma_b(v)$

Figure 3: Average and standard deviation of the mean absolute errors of $\hat{\sigma}_a(v)$ and $\sigma_b(v)$ for each network.

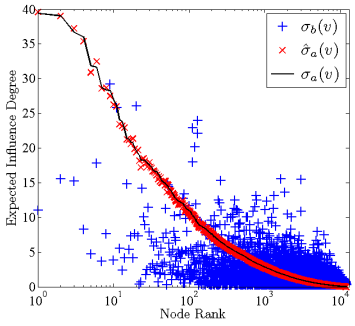
larger as c increases[8]. Especially, the difference between σ_a and σ_b is quite large for the top 100 nodes for $c = 0.5$ and $c = 1.0$. Note that the horizontal axis is logarithmic scale. This implies that for a majority of nodes the difference between $\sigma_a(v)$ and $\sigma_b(v)$ is very small. This explains why the mean absolute error is low as shown in Fig.3. We observed the same tendency for the other three networks, and here show only the results for the Coauthorship network in Fig. 5 for a reference. In summary, these results indicate that it is important to obtain the parameter values accurately in order to estimate the influence degree of each node in good accuracy.

In case of the basic model, it has been proven that the influence degree $\sigma_b(v)$ of a node is proportional to the degree of the node (Section 2.1). We explored to find a measure that correlates well to the influence degree in case of AwVM. Figure 6 shows the relation between the node degree and the influence degree (nodes ranked according to the degree) and Fig. 7 the relation between the node strength and the influence degree (nodes ranked accordint to the strength), both for the same trial for the blog network as in Fig. 4. The black solid line $\bar{d}(v)$ in Fig. 6 represents the value of $d(v)$ that is normalized so that the total sum of $d(v)$ over all nodes becomes equal to the total sum of the

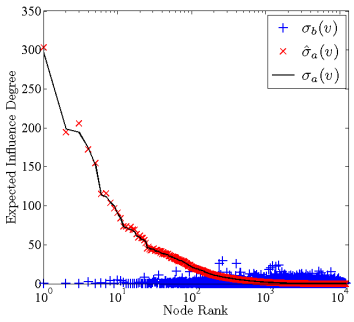
8. Note that the scale of the vertical axis of Fig. 4c is much larger than the other two figures.



(a) Non-uniformity parameter c = 0.1

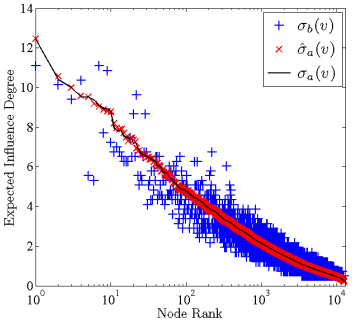


(b) Non-uniformity parameter c = 0.5

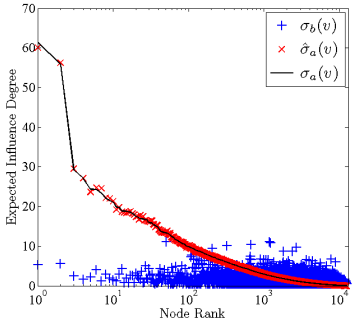


(c) Non-uniformity parameter c = 1.0

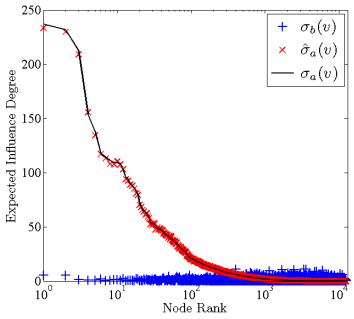
Figure 4: Comparison of influence degree, $\sigma_a(v)$, $\hat{\sigma}_a(v)$, and $\sigma_b(v)$ for the blog network for one particular run, randomly selected from the 10 independent trials.



(a) Non-uniformity parameter c = 0.1



(b) Non-uniformity parameter c = 0.5



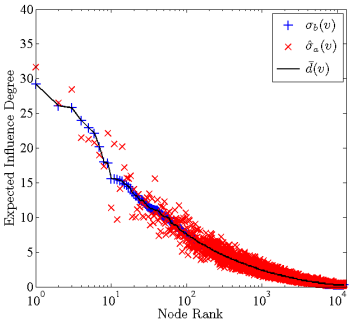
(c) Non-uniformity parameter c = 1.0

Figure 5: Comparison of influence degree, $\sigma_a(v)$, $\hat{\sigma}_a(v)$, and $\sigma_b(v)$ for the Coauthorship network for one particular run, randomly selected from the 10 independent trials.

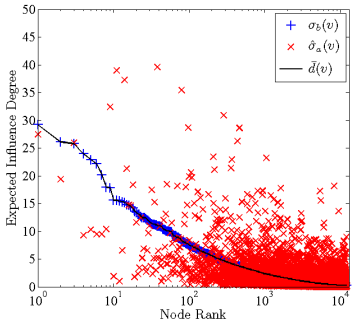
influence degree, and the black solid line $\bar{s}(v)$ in Fig. 7 represents the value of s_v that is normalized in the same way. From Fig. 6, we can see that the node degree $d(v)$ can be a good estimator of $\hat{\sigma}_a(v)$ for $c = 0.1$, but it does not work well for a larger value of c. This is because the AwVM becomes closer to the basic voter model as the parameter c becomes closer to 0. Indeed, in Fig. 6, σ_b overlaps with the curve that represents $\bar{d}(v)$ because the naive model with $s_v = 1.0$, *i.e.*, $c = 0$, is identical to the basic voter model. Conversely, Fig. 7 shows that s_v can be a good estimator of $\hat{\sigma}_a(v)$ in case of $c = 1.0$, but it does not work well for a smaller c. Also in this case, the node degree $\bar{d}(v) = \sigma_b(v)$ can approximate $\hat{\sigma}_a(v)$ in good accuracy only for $c = 0.1$[9].

These results suggest that although either the node strength s_v or the node degree $d(v)$ alone cannot be a good estimator of the influence degree of the AwVM, their combination could be a good estimator of the influence degree. To confirm this hypothesis, we further investigated the relation between their product, *i.e.*, $g(v) = s_v d(v)$, which is referred to as the *strength-weight degree*, and the influence degree for the AwVM. The result is shown in Fig. 8, where the black solid line $\bar{g}(v)$ repre-

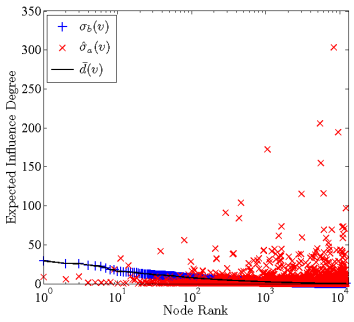
9. It does not look so at first glance, but note that the number of plots are the same for both $\sigma_b(v)$ and $\hat{\sigma}_a(v)$ and there are many overlaps.



(a) Non-uniformity parameter $c = 0.1$

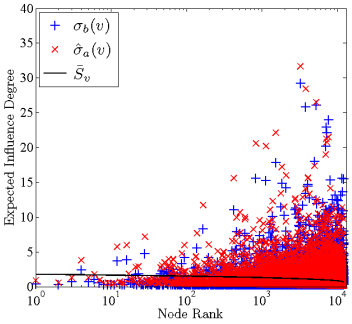


(b) Non-uniformity parameter $c = 0.5$

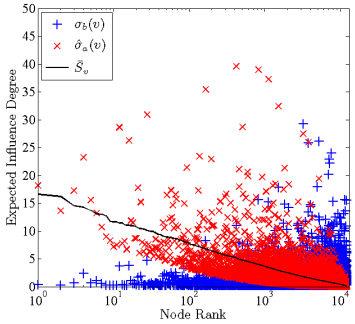


(c) Non-uniformity parameter $c = 1.0$

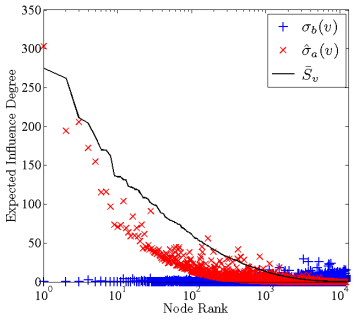
Figure 6: Relation between the node degree $d(v)$ and the influence degree, ($\hat{\sigma}_a(v)$ and $\sigma_b(v)$) for the blog network for one particular run, randomly selected from the 10 independent trials.



(a) Non-uniformity parameter $c = 0.1$



(b) Non-uniformity parameter $c = 0.5$

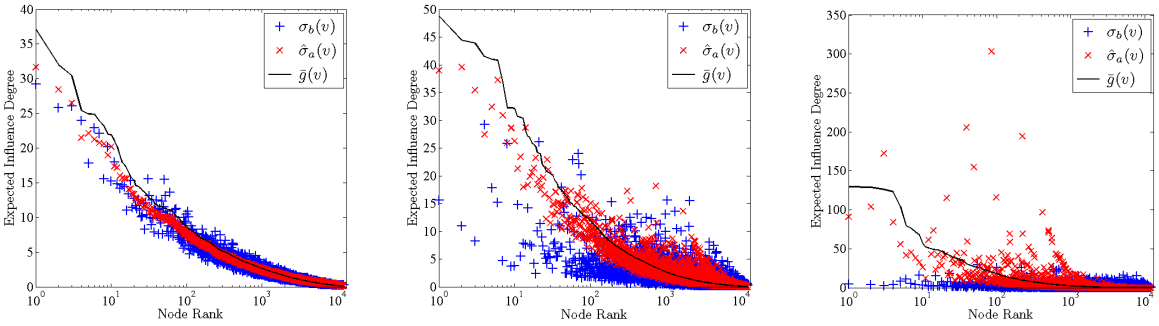


(c) Non-uniformity parameter $c = 1.0$

Figure 7: Relation between the node strength s_v and the influence degree ($\hat{\sigma}_a(v)$ and $\sigma_b(v)$) for the blog network for one particular run, randomly selected from the 10 independent trials.

sents the value of $g(v)$ that is normalized in the same way as $\bar{d}(v)$ in Fig. 6 (nodes ranked according to $g(v)$). Again, we used the results that are obtained from the same trial for the blog network as in Fig. 4. Unfortunately, this result refutes the aforementioned hypothesis. It can approximate the influence degree well only for $c = 0.1$ as $d(v)$ does, but not for a larger c. We examined the other three networks and found that the results strongly depend on the sparsness of the network. Wikipedia and Ameblo networks are much denser than Blog network. They showed better results for a larger c. We then randomely deleted links from these two networks and confirmed that the results become worse as more links are deleted. The reason behind this needs further investigation.

From these results, we can conclude that the opinion formation process for the AwVM becomes quite complex due to the introduction of the node strength, which is not the case for the basic voter model, and it seems that there is no simple heuristic measure that is able to estimate the influence degree in good accuracy for a wide range of networks. The *strength-weight degree* can be a good measure in some cases but not in all.



(a) $c = 0.1$ (b) $c = 0.5$ (c) $c = 1.0$

Figure 8: Relation between the product of the node strength and the node degree ($g(v) = s_v d(v)$) and the influence degree ($\hat{\sigma}_a$ and σ_b) for the blog network for one particular run, randomly selected from the 10 independent trials.

4.5 Discussion

We discuss the relation of the AwVM to the existing two models mentioned earlier. Even-Dar and Shapria (2007) investigated the influence maximization problem based on the basic voter model. They assumed that there need some initial costs to make each node accept an initial opinion. This cost is completely different from the node strength argued in this paper in that the initial cost of each node does not affect its influence degree at all. In contrast, the node strength directly affects the influence degree of each node, as shown in our experiments. Clearly, it is straightforward to incorporate initial costs to the AwVM. The theoretical analysis for the VwVM revealed that the opinion with the highest value wins and all the others die (winner-take-all process) for a situation where the local opinion share can be approximated by the average opinion share over the whole network, (e.g., the case of a complete network) (Kimura et al., 2010b). There does not exist an effective formula of calculating the influence degree for the VwVM, which corresponds to Eq. (3) for the AwVM, and obtaining the influence degree for the VwVM is computationally expensive. Clearly, node strength and opinion value are different concept, and it is possible to introduce the opinion values to the AwVM. However, how the node strength affects the winner-take-all process remains as an open question.

In this paper, we assumed that the network is static. Dynamic social network is easily handled by the current method by updating the neighbor nodes right before time t, incorporating the newly added/deleted nodes and updating the neighbors opinion distribution $n_k(t, v)$ accordingly. The same algorithm runs without any other changes.

5. Conclusion

Opinion formation over a social network was analyzed by modeling the cascade of interactions of neighboring nodes as probabilistic process of state changes. We modeled this process as a variant of the well known voter model with emphasis given on the strength of each node, called Attribute-weighted Voter Model with Multiple Opinions (AwVM). The strength reflects the degree of direct influence of the node, and we addressed the problem of estimating this strength from the observed opinion formation data. As each node has its own strength, the number of variables we want to

estimate is as large as the number of nodes in the network, which requires a prohibitively large amount of training data. We avoided this problem by assuming a functional dependency of the node strength on the small number of selected node attributes, which we believe to reflect the reality, and learn the parameter values that specify the functional dependency without a need for such a large amount of data. The task was formulated as the maximum likelihood estimation problem, and an efficient parameter value update algorithm that guarantees the convergence was derived. We evaluated the performance of the learning algorithm on four real world networks assuming a particular class of attribute dependency, and confirmed that the dependency can be correctly learned. We further showed that the influence degree of each node (expected number of the nodes at the end of opinion formation process that have the same opinion as the starting node considered) based on our AwVM is substantially different from that obtained assuming a model with uniform strength, *i.e.* without the strength, and the sensitivity analysis indicated that the influence degree is more sensitive to the node strength than the node degree even for a moderate value of the node strength. The results, at first glance, suggested to use the strength-weight degree as a rough measure of approximating influence degree, but quantitative evaluation refuted this hypothesis. Introduction of node strength, which is quite natural and reflects the reality, brings complexity to the opinion formation process, and there appears to be no simple heuristic measure that can predict the influence degree in good accuracy for a wide range of networks.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 23500312).

References

C. Castellano, M. A. Munoz, and R. Pastor-Satorras. Nonlinear *q*-voter model. *Physical Review E*, 80:041129, 2009.

D. Crandall, D. Cosley, D. Huttenlocner, J. Kleinberg, and S. Suri. Feedback effects between similarity and sociai infiuence in online communities. In *Proceedings of KDD 2008*, pages 160–168, 2008.

P. Domingos. Mining social networks for viral marketing. *IEEE Intell. Syst.*, 20:80–82, 2005.

E. Even-Dar and A. Shapria. A note on maximizing the spread of influence in social networks. In *Proceedings of WINE 2007*, pages 281–286, 2007.

J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12:211–223, 2001.

D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. *SIGKDD Explorations*, 6:43–52, 2004.

P. Holme and M. E. J. Newman. Nonequilibrium phase transition in the coevolution of networks and opinions. *Physical Review E*, 74:056108, 2006.

D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD-2003*, pages 137–146, 2003.

M. Kimura, K. Saito, and H. Motoda. Minimizing the spread of contamination by blocking links in a network. In *AAAI-08*, pages 1175–1180, 2008.

M. Kimura, K. Saito, and H. Motoda. Blocking links to minimize contamination spread in a social network. *ACM Trans. Knowl. Discov. Data*, 3:9:1–9:23, 2009.

M. Kimura, K. Saito, and R. Nakano. Extracting influential nodes for information diffusion on a social network. In *AAAI-07*, pages 1371–1376, 2007.

M. Kimura, K. Saito, R. Nakano, and H. Motoda. Extracting influential nodes on a social network for information diffusion. *Data Min. and Knowl. Disc.*, 20:70–97, 2010a.

M. Kimura, K. Saito, K. Ohara, and H. Motoda. Learning to predict opinion share in social networks. In *AAAI-10*, pages 1364–1370, 2010b.

J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. In *EC'06*, pages 228–237, 2006.

T. M. Liggett. *Stochastic interacting systems: contact, voter, and exclusion processes*. Spriger, New York, 1999.

M. E. J. Newman. The structure and function of complex networks. *SIAM Rev.*, 45:167–256, 2003.

M. E. J. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. *Phys. Rev. E*, 66:035101, 2002.

G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

K. Saito, M. Kimura, K. Ohara, and H. Motoda. Detecting changes in opinion value distribution for voter model. In *Proceedings of International Conference on Social Computing, Behavioral-Cultural Modeling, & Prediction (SBP11)*, pages 89–96. Spriger, LNAI 6589, 2011.

V. Sood and S. Redner. Voter model on heterogeneous graphs. *Physical Review Letters*, 94:178701, 2005.

H. Tong, B. A. Prakash, C. Tsoourakakis, T. Eliassi-Rad, C. Faloutsos, and D. H. Chau. On the vulnerability of large graphs. In *ICDM 2010*, pages 1091–1096, 2010.

D. J. Watts. A simple model of global cascades on random networks. *PNAS*, 99:5766–5771, 2002.

D. J. Watts and P. S. Dodds. Influence, networks, and public opinion formation. *J. Cons. Res.*, 34: 441–458, 2007.

H. Yang, Z.. Wu, C. Zhou, T. Zhou, and B. Wang. Effects of social diversity on the emergence of global consensus in opinion dynamics. *Physical Review E*, 80:046108, 2009.

Detecting Anti-majority Opinionists Using Value-weighted Mixture Voter Model

Masahiro Kimura[1], Kazumi Saito[2], Kouzou Ohara[3], and Hiroshi Motoda[4]

[1] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[2] School of Administration and Informatics, University of Shizuoka
Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[3] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We address the problem of detecting anti-majority opinionists using the value-weighted mixture voter (VwMV) model. This problem is motivated by the fact that some people do not always agree with the majority and support the minority. We extend the value-weighted voter model to include this phenomenon with the anti-majoritarian tendency of each node as a new parameter, and learn this parameter as well as the value of each opinion from a sequence of observed opinion data over a social network. We experimentally show that it is possible to learn the anti-majoritarian tendency of each node correctly as well as the opinion values, whereas a naive approach which is based on a simple counting heuristic fails. We also show theoretically that, in a situation where the local opinion share can be approximated by the average opinion share, it is not necessarily the case that the opinion with the highest value prevails and wins when the opinion values are non-uniform, whereas the opinion share prediction problem becomes ill-defined and any opinion can win when the opinion values are uniform. The simulation results support that this holds for typical real world social networks.

1 Introduction

The emergence of large scale social computing applications has made massive social network data available as well as our daily life much depend on these networks through which news, ideas, opinions and rumors can spread [17, 16, 7, 5]. Thus, investigating the spread of influence in social networks has been the focus of attention [14, 4, 20]. The most well studied problem would be the *influence maximization problem*, that is, the problem of finding a limited number of influential nodes that are effective for spreading information. Many new algorithms that can effectively find approximate solutions have been proposed both for estimating the expected influence and for finding good candidate nodes [9, 11, 15, 2, 3]. However, the models used above allow a node in the network to

take only one of the two states, *i.e.*, either active or inactive, because the focus is on *influence*.

Application such as an on-line competitive service in which a user can choose one from multiple choices and decisions requires a different approach where a model must handle multiple states. Also important is to consider the value of each choice, *e.g.*, quality, brand, authority, etc., because this impacts other's choice. Opinion formation and its spread fit in the same class of problems. The model best suited for this kind of analysis would be a voter model [19, 8, 6, 4, 1, 21], which is one of the most basic stochastic process model and has the same key property with the *linear threshold model* used in information diffusion that a node decision is influenced by its neighbor's decision, *i.e.*, a person changes its opinion by the opinions of its neighbors. In [12], we extended the voter model to include the opinion values, and addressed the problem of predicting the opinion share at a future time by learning the opinion values from a limited amount of past observed opinion diffusion data. Interestingly, theoretical analysis for a situation where the local opinion share can be approximated by the average opinion share over the whole network, (*e.g.*, the case of a complete network), revealed that the expected share prediction problem is well-defined only when the opinion values are non-uniform, in which case the final consensus is winner-take-all, *i.e.*, the opinion with the highest value wins and all the others die, and when they are uniform, any opinion can be a winner.

The problem we address in this paper challenges the same problem, but from a different angle. In the voter model including its variants, it is assumed that people naturally tend to follow their neighbors' majority opinion. However, we note that there are always people who do not agree with the majority and support the minority's opinion. We are interested in how this affects the opinion share, and have extended the value-weighted voter model with multiple opinions to include this anti-majority effect with the *anti-majoritarian tendency* of each node as a new parameter. We are not the first to introduce the notion of anti-majority. There is a model called anti-voter model where only two opinions are considered. Each one chooses one of its neighbors randomly and decides to take the opposite opinion of the neighbor chosen. Röllin [18] analyzed the statistical property of the anti-voter model introducing the notion of exchangeable pair couplings. We have extended the simple anti-voter model to value-weighted anti-voter model with multiple opinions, and combined it linearly with the value-weighted voter model with multiple opinions. The model now has a new parameter at each node which is a measure for the the anti-majoritarian tendency (weight for the value-weighted anti-voter model) in addition to the original parameter (opinion value), and we call the combined model the *value-weighted mixture voter (VwMV) model*.

Both the parameters, anti-majoritarian tendency and opinion value, can be efficiently learned by an iterative algorithm (EM algorithm) that maximizes the likelihood of the model's generating the observed data. We tested the algorithm for three real world social networks with the size ranging over 4,000 to 10,000 nodes and 40,000 to 250,000 links, and experimentally showed that the parameter value update algorithm correctly identifies the anti-majoritarian tendency of each node under various situations provided that there are enough data. The anti-majoritarian tendency estimated by using a heuristic that simply counts the number of opinion updates in which the chosen opinion is the

same as the minority's opinion turns out to be a very poor approximation. These results show that the model learned by the proposed algorithm can be used to predict the future opinion share and provides a way to analyze such problems as influence maximization or minimization for opinion diffusion under the presence of anti-majority opinionists. Similar analysis as in [12] revealed interesting results for the average behavior that the opinion share crucially depends on the anti-majoritarian tendency and that the opinion with the highest value does not necessarily prevail when the values are non-uniform, which is in contrast to the result of the value-weighted voter model, whereas the share prediction problem becomes ill-defined when the opinion values are uniform, *i.e.*, any opinion can win, which is the same as in the value-weighted voter model. The simulation results also support that this holds for typical real world social networks.

2 Opinion Dynamics Models

We define the VwMV model. Let $G = (V, E)$ be an undirected (bidirectional) network with self-loops, where V and E ($\subset V \times V$) are the sets of all the nodes and links in the network, respectively. For a node $v \in V$, let $\Gamma(v)$ denote the set of neighbors of v in G, that is,

$$\Gamma(v) = \{u \in V;\ (u, v) \in E\}.$$

Note that $v \in \Gamma(v)$. Given an integer K with $K \geq 2$, we consider the spread of K opinions (opinion $1, \cdots$, opinion K) on G, where each node holds one of the K opinions at any time t (≥ 0). We assume that each node of G initially holds one of the K opinions with equal probability at time $t = 0$. Let $f_t : V \to \{1, \cdots, K\}$ denote the *opinion distribution* at time t, where $f_t(v)$ stands for the opinion of node v at time t. Note that f_0 stands for the initial opinion distribution. For any $v \in V$ and $k \in \{1, 2, \cdots, K\}$, let $n_k(t, v)$ be the number of v's neighbors that hold opinion k as the latest opinion before time t, *i.e.*,

$$n_k(t, v) = |\{u \in \Gamma(v);\ \varphi_t(u) = k\}|,$$

where $\varphi_t(u)$ is the latest opinion of u before time t.

2.1 Voter and Anti-voter Models

We revisit the voter model, which is one of the standard models of opinion dynamics, where K is usually set to 2. The evolution process of the voter model is defined as follows:

1. At time 0, each node v independently decides its update time t according to some probability distribution such as an exponential distribution with parameter 1.[1] The successive update time is determined similarly at each update time t.
2. At update time t, the node v adopts the opinion of a randomly chosen neighbor u, *i.e.*,

$$f_t(v) = \varphi_t(u).$$

[1] This assumes that the average delay time is 1.

3. The process is repeated from the initial time $t = 0$ until the next update-time pasts a given final-time T.

We note that in the voter model each individual tends to adopt the majority opinion among its neighbors. Thus, we can extend the original voter model with 2 opinions to a voter model with K opinions by replacing Step 2 with: At update time t, the node v selects one of the K opinions according to the probability distribution,

$$P(f_t(v) = k) = \frac{n_k(t, v)}{|\Gamma(v)|}, \quad (k = 1, \cdots, K). \tag{1}$$

The anti-voter model is defined in the similar way. In this model K is set to 2 and Step 2 is replaced with: At update time t, the node v adopts the opposite opinion of a randomly chosen neighbor u, *i.e.*,

$$f_t(v) = 3 - \varphi_t(u).$$

We note that each individual tends to adopt the minority opinion among its neighbors instead. The anti-voter model with K opinions is obtained by replacing Eq. (1) with

$$P(f_t(v) = k) = \frac{1}{K-1}\left(1 - \frac{n_k(t, v)}{|\Gamma(v)|}\right), \quad (k = 1, \cdots, K). \tag{2}$$

2.2 Value-weighted Mixture Voter Model

In order to investigate the competitive spread of K opinions, it is important to consider each opinion's value because this impacts other's choice. In [12], we extended the voter model with K opinions to the *value-weighted voter model* by introducing the parameter (*opinion value* of opinion k) w_k (> 0). In this model, Eq. (1) was replaced with

$$P(f_t(v) = k) = p_k(t, v, \boldsymbol{w}), \quad (k = 1, \cdots, K),$$

where $\boldsymbol{w} = (w_1, \cdots, w_K)$ and

$$p_k(t, v, \boldsymbol{w}) = \frac{w_k\, n_k(t, v)}{\sum_{j=1}^{K} w_j\, n_j(t, v)}, \quad (k = 1, \cdots, K). \tag{3}$$

We can also extend the anti-voter model with K opinions to the *value-weighted anti-voter model* by replacing Eq. (2) with

$$P(f_t(v) = k) = \frac{1 - p_k(t, v, \boldsymbol{w})}{K-1}, \quad (k = 1, \cdots, K). \tag{4}$$

Further, we can define the *value-weighted mixture voter (VwMV) model* by replacing Eq. (4) with

$$P(f_t(v) = k) = (1 - \alpha_v)\, p_k(t, v, \boldsymbol{w}) + \alpha_v\, \frac{1 - p_k(t, v, \boldsymbol{w})}{K-1}, \quad (k = 1, \cdots, K), \tag{5}$$

where α_v is a parameter with $0 \leq \alpha_v \leq 1$. Note that each individual located at node v tends to behave like a majority opinionist if the value of α_v is small, and tends to behave like an anti-majority opinionist if the value of α_v is large. Therefore, we refer to α_v as the *anti-majoritarian tendency* of node v.

3 Learning Problem and Behavior Analysis

We consider the problem of identifying the VwMV model on network G from observed data $\mathcal{D}_T$ in time-span $[0, T]$, where $\mathcal{D}_T$ consists of a sequence of (v, t, k) such that node v changed its opinion to opinion k at time t for $0 \leq t \leq T$. The identified model can be used to predict how much of the share each opinion will have at a future time T' $(> T)$, and to identify both high anti-majoritarian tendency nodes (*i.e.*, anti-majority opinionists) and low anti-majoritarian tendency nodes (*i.e.*, majority opinionists). Below, we theoretically investigate some basic properties of the VwMV model, and demonstrate that it is crucial to accurately estimate the values of the parameters, w_k, $(k = 1, \cdots, K)$ and α_v, $(v \in V)$.

For any opinion k, let $h_k(t)$ denote its *population* at time t, *i.e.*,

$$h_k(t) = |\{v \in V;\ f_t(v) = k\}|,$$

and let $g_k(t)$ denote its expected *share* at time t, *i.e.*,

$$g_k(t) = \left\langle \frac{h_k(t)}{\sum_{j=1}^{K} h_j(t)} \right\rangle.$$

We investigate the behavior of expected share $g_k(t)$ for a sufficiently large t. According to the previous work in statistical physics (*e.g.*, [19]), we employ a mean field approach. We first consider a rate equation,

$$\frac{dg_k(t)}{dt} = (1 - g_k(t))\, P_k(t) - g_k(t)\,(1 - P_k(t)), \quad (k = 1, \cdots, K), \tag{6}$$

where $P_k(t)$ denotes the probability that a node adopts opinion k at time t. Note that in the right-hand side of Eq. (6), $g_k(t)$ is regarded as the probability of choosing a node holding opinion k at time t. Here, we assume that the average local opinion share $\langle n_k(t, v) / \sum_{j=1}^{K} n_j(t, v) \rangle$ in the neighborhood of a node v can be approximated by the expected opinion share $g_k(t)$ of the whole network for each opinion k. Then, we obtain the following approximation from Eq. (5):

$$P_k(t) = (1 - \alpha)\, \tilde{p}_k(t, \boldsymbol{w}) + \alpha\, \frac{1 - \tilde{p}_k(t, \boldsymbol{w})}{K - 1}, \quad (k = 1, \cdots, K), \tag{7}$$

where α is the average value of anti-majoritarian tendency α_v, $(v \in V)$, and

$$\tilde{p}_k(t, \boldsymbol{w}) = \frac{w_k\, g_k(t)}{\sum_{j=1}^{K} w_j\, g_j(t)}, \quad (k = 1, \cdots, K). \tag{8}$$

Note that Eqs. (7) and (8) are exactly satisfied when G is a complete network and the anti-majoritarian tendency is node independent, *i.e.*, $\alpha_v = \alpha$, $(\forall v \in V)$.

For the value-weighted voter model (*i.e.*, $\alpha = 0$), we theoretically showed the following results in [12]:

1. When the opinion values are uniform (*i.e.*, $w_1 = \cdots = w_K$), any opinion can become a winner, that is, if $g_1(0) = \cdots = g_K(0) = 1/K$, then $g_k(t) = 1/K$, $(t \geq 0)$ for each opinion k.
2. When the opinion values are non-uniform, the opinion k^* with highest opinion value is expected to finally prevail over the others, that is, $\lim_{t \to \infty} g_{k^*}(t) = 1$.

We extend these results to the VwMV model below.

Case of uniform opinion values: We suppose that $w_1 = \cdots = w_K$. Then, since $\sum_{k=1}^{K} g_k(t) = 1$, from Eq. (8), we obtain

$$\tilde{p}_k(t, \boldsymbol{w}) = g_k(t), \quad (k = 1, \cdots, K).$$

Thus, we can easily derive from Eqs. (6) and (7) that

$$\frac{dg_k(t)}{dt} = -\frac{\alpha}{1 - 1/K} \left(g_k(t) - \frac{1}{K} \right), \quad (k = 1, \cdots, K).$$

Hence, we have

$$\lim_{t \to \infty} g_k(t) = 1/K, \quad (k = 1, \cdots, K).$$

Case of non-uniform opinion values: We assume that the opinion values are non-uniform. We parameterize the non-uniformity by the ratio,

$$s_k = \frac{w_k}{\sum_{j=1}^{K} w_j / K}, \quad (k = 1, \cdots, K).$$

Let k^* be the opinion with the highest value parameter. Note that $s_{k^*} > 1$. We assume for simplicity that

$$w_k = w' \ (< w_{k^*}) \quad \text{if } k \neq k^*,$$

where w' is a positive constant. We also assume that

$$g_1(0) = \cdots = g_K(0) = 1/K.$$

We can see from the symmetry of the setting that $g_k(t) = g_\ell(t)$, $(t \geq 0)$ if $k, \ell \neq k^*$. This implies that opinion k^* is the winner at time t if and only if $g_{k^*}(t) > 1/K$. Here, suppose that there exists some time $t_0 > 0$ such that

$$g_{k^*}(t_0) = 1/K.$$

Then, from Eqs. (6) and (8), we obtain

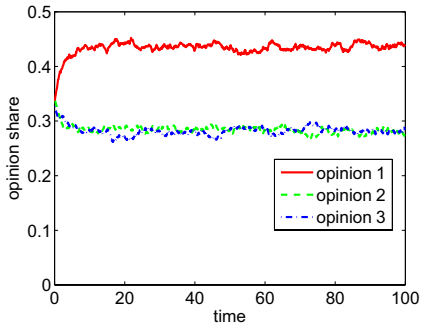
$$\left. \frac{dg_{k^*}(t)}{dt} \right|_{t=t_0} = P_{k*}(t_0) - \frac{1}{K}, \qquad \tilde{p}_{k^*}(t_0, \boldsymbol{w}) = \frac{s_{k^*}}{K}.$$

Thus we have from Eq. (7) that

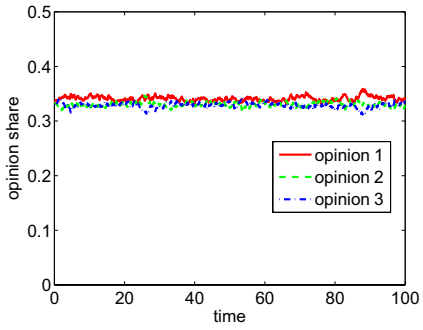
$$\left. \frac{dg_{k^*}(t)}{dt} \right|_{t=t_0} = \frac{s_{k^*} - 1}{K - 1} \left(1 - \frac{1}{K} - \alpha \right).$$

Therefore, we obtain the following results:

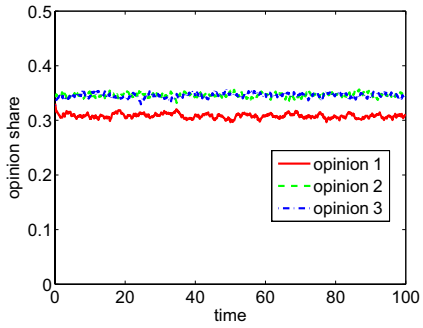
1. When $\alpha < 1 - 1/K$,
$$g_{k^*}(t) > 1/K, \quad (t > 0),$$
that is, opinion k^* is expected to spread most widely and become the majority.
2. When $\alpha = 1 - 1/K$,
$$g_k(t) = 1/K, \quad (t \geq 0),$$
for any opinion k, that is, any opinion can become a winner.
3. When $\alpha > 1 - 1/K$,
$$g_{k^*}(t) < 1/K, \quad (t > 0),$$
that is, opinion k^* is expected to spread least widely and become the minority.



(a) $a = 2$, $b = 4$, ($\alpha < 1 - 1/3$)



(b) $a = 4$, $b = 2$, ($\alpha = 1 - 1/3$)



(c) $a = 18$, $b = 2$, ($\alpha > 1 - 1/3$)

Fig. 1: Results of the opinion share curves for different distributions of anti-majoritarian tendency in the blog network.

Experiments: The above theoretical results are justified only when the approximation (see Eq. (7)) holds, which is always true in the case of complete networks. Real social networks are much more sparse and thus, we need to verify to how much extent the above results are true for real networks. We experimentally confirmed the above theoretical results for several real-world networks. Here, we present the experimental results for $K = 3$ in the Blog network (see the section of "Experimental Evaluation" below), where the opinion values are $w_1 = 2$, $w_2 = w_3 = 1$, and anti-majoritarian tendency α_v, ($v \in V$) is drawn from the beta distribution with shape parameters a and b. Figure 1 shows the results of opinion share curves, $t \mapsto h_k(t) \Big/ \sum_{j=1}^{K} h_j(t)$, ($k = 1, 2, 3$), when the distribution of anti-majoritarian tendency changes, where each node adopted one of three opinions with equal probability at time $t = 0$. Note that $\alpha = 0.33 (< 1 - 1/3)$ if $a = 2$, $b = 4$, $\alpha = 1 - 1/3$ if $a = 4$, $b = 2$, and $\alpha = 0.9 (> 1 - 1/3)$ if $a = 18$, $b = 2$. These results support the validity of our theoretical analysis. We also note that the results shown in Figures 1a, 1b and 1c were qualitatively universal although the opinion dynamics is stochastic.

4 Learning Method

We describe a method for estimating parameter values of the VwMV model from given observed opinion spreading data $\mathcal{D}_T$. Based on the evolution process of our model (see Eq. (5)), we can obtain the likelihood function,

$$\mathcal{L}(\mathcal{D}_T; \boldsymbol{w}, \boldsymbol{\alpha}) = \log\left(\prod_{(v,t,k)\in\mathcal{D}_T} P(f_t(v) = k)\right), \tag{9}$$

where $\boldsymbol{w}$ stands for the K-dimensional vector of value parameters, *i.e.*, $\boldsymbol{w} = (w_1, \cdots, w_K)$, and $\boldsymbol{\alpha}$ is the $|V|$-dimensional vector with each element α_v being the anti-majoritarian tendency of node v. Thus our estimation problem is formulated as a maximization problem of the objective function $\mathcal{L}(\mathcal{D}_T; \boldsymbol{w}, \boldsymbol{\alpha})$ with respect to $\boldsymbol{w}$ and $\boldsymbol{\alpha}$. Note from Eqs. (3), (5) and (9) that $\mathcal{L}(\mathcal{D}_T; c\boldsymbol{w}, \boldsymbol{\alpha}) = c\mathcal{L}(\mathcal{D}_T; \boldsymbol{w}, \boldsymbol{\alpha})$ for any $c > 0$. Note also that each opinion value w_k is positive. Thus, we transform the parameter vector $\boldsymbol{w}$ by $\boldsymbol{w} = \boldsymbol{w}(\boldsymbol{z})$, where

$$\boldsymbol{w}(\boldsymbol{z}) = (e^{z_1}, \cdots, e^{z_{K-1}}, 1), \quad \left(\boldsymbol{z} = (z_1, \cdots, z_{K-1}) \in \mathbf{R}^{K-1}\right). \tag{10}$$

Namely, our problem is to estimate the values of $\boldsymbol{z}$ and $\boldsymbol{\alpha}$ that maximize $\mathcal{L}(\mathcal{D}_T; \boldsymbol{w}(\boldsymbol{z}), \boldsymbol{\alpha})$.

We derive an EM like iterative algorithm for obtaining the maximum likelihood estimators. To this purpose, we introduce the following parameters that depends on $\boldsymbol{\alpha}$: For any $v \in V$ and $k, j \in \{1, \cdots, K\}$,

$$\beta_{v,k,j}(\boldsymbol{\alpha}) = \begin{cases} 1 - \alpha_v & \text{if} \quad j = k, \\ \alpha_v/(K-1) & \text{if} \quad j \neq k. \end{cases} \tag{11}$$

Then, from the definition of $P(f_t(v) = k)$ (see Eq. (5)), by noting $1 - p_k(t, v, \boldsymbol{w}) = \sum_{j\neq k} p_j(t, v, \boldsymbol{w})$, we can express Eq. (9) as follows:

$$\mathcal{L}(\mathcal{D}_T; \boldsymbol{w}(\boldsymbol{z}), \boldsymbol{\alpha}) = \sum_{(v,t,k)\in\mathcal{D}_T} \log\left(\sum_{j=1}^{K} \beta_{v,k,j}(\boldsymbol{\alpha})\ p_j(t, v, \boldsymbol{w}(\boldsymbol{z}))\right).$$

Now, let $\overline{\boldsymbol{z}}$ and $\overline{\boldsymbol{\alpha}}$ be the current estimates of $\boldsymbol{z}$ and $\boldsymbol{\alpha}$, respectively. Then, by considering the posterior probabilities,

$$q_{v,t,k,j}(\boldsymbol{z}, \boldsymbol{\alpha}) = \frac{\beta_{v,k,j}(\boldsymbol{\alpha})\ p_j(t, v, \boldsymbol{w}(\boldsymbol{z}))}{\sum_{i=1}^{K} \beta_{v,k,i}(\boldsymbol{\alpha})\ p_i(t, v, \boldsymbol{w}(\boldsymbol{z}))},$$

($v \in V$, $0 \leq t \leq T$ $k, j = 1, \cdots, K$), we can transform our objective function as follows:

$$\mathcal{L}(\mathcal{D}_T; \boldsymbol{w}(\boldsymbol{z}), \boldsymbol{\alpha}) = Q(\boldsymbol{z}, \boldsymbol{\alpha}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}) - \mathcal{H}(\boldsymbol{z}, \boldsymbol{\alpha}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}), \tag{12}$$

where $Q(\boldsymbol{z}, \boldsymbol{\alpha}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})$ is defined by

$$Q(\boldsymbol{z}, \boldsymbol{\alpha}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}) = Q_1(\boldsymbol{z}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}) + Q_2(\boldsymbol{\alpha}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}), \tag{13}$$

$$Q_1(\boldsymbol{z}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}) = \sum_{(v,t,k)\in\mathcal{D}_T} \sum_{j=1}^{K} q_{v,t,k,j}(\overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})\ \log p_j(t, v, \boldsymbol{w}(\boldsymbol{z})), \tag{14}$$

$$Q_2(\boldsymbol{\alpha}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}) = \sum_{(v,t,k)\in\mathcal{D}_T} \sum_{j=1}^{K} q_{v,t,k,j}(\overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})\ \log \beta_{v,k,j}(\boldsymbol{\alpha}), \tag{15}$$

and $\mathcal{H}(\boldsymbol{z}, \boldsymbol{\alpha}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})$ is defined by

$$\mathcal{H}(\boldsymbol{z}, \boldsymbol{\alpha}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}) = \sum_{(v,t,k)\in\mathcal{D}_T} \sum_{j=1}^{K} q_{v,t,k,j}(\overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}) \log q_{v,t,k,j}(\boldsymbol{z}, \boldsymbol{\alpha}).$$

Since $\mathcal{H}(\boldsymbol{z}, \boldsymbol{\alpha}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})$ is maximized at $\boldsymbol{z} = \overline{\boldsymbol{z}}$ and $\boldsymbol{\alpha} = \overline{\boldsymbol{\alpha}}$, we can increase the value of $\mathcal{L}(\mathcal{D}_T; \boldsymbol{w}(\boldsymbol{z}), \boldsymbol{\alpha})$ by maximizing $\mathcal{Q}(\boldsymbol{z}, \boldsymbol{\alpha}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})$ with respect to $\boldsymbol{z}$ and $\boldsymbol{\alpha}$ (see Eq. (12)). From Eq. (13), we can maximize $\mathcal{Q}(\boldsymbol{z}, \boldsymbol{\alpha}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})$ by independently maximizing $\mathcal{Q}_1(\boldsymbol{z}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})$ and $\mathcal{Q}_2(\boldsymbol{\alpha}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})$ with respect to $\boldsymbol{z}$ and $\boldsymbol{\alpha}$, respectively.

First, we estimate the value of $\boldsymbol{z}$ that maximizes $\mathcal{Q}_1(\boldsymbol{z}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})$. Here, note from Eqs.(3) and (10) that for $j = 1, \cdots, K$ and $\lambda = 1, \cdots, K-1$,

$$\frac{\partial p_j(t, v, \boldsymbol{w}(\boldsymbol{z}))}{\partial z_\lambda} = \delta_{j,\lambda}\, p_j(t, v, \boldsymbol{w}(\boldsymbol{z})) - p_j(t, v, \boldsymbol{w}(\boldsymbol{z}))\, p_\lambda(t, v, \boldsymbol{w}(\boldsymbol{z})), \tag{16}$$

where $\delta_{j,\lambda}$ is the Kronecker's delta. From Eqs. (14) and (16), we have

$$\frac{\partial Q_1(\boldsymbol{z}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})}{\partial z_\lambda} = \sum_{(v,t,k)\in\mathcal{D}_T} \sum_{j=1}^{K} q_{v,t,k,j}(\overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}) \left(\delta_{j,\lambda} - p_\lambda(t, v, \boldsymbol{w}(\boldsymbol{z}))\right), \tag{17}$$

for $\lambda = 1, \cdots, K-1$. Moreover, from Eqs. (16) and (17), we have

$$\frac{\partial^2 Q_1(\boldsymbol{z}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})}{\partial z_\lambda\, \partial z_\mu} = \sum_{(v,t,k)\in\mathcal{D}_T} \sum_{j=1}^{K} q_{v,t,k,j}(\overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}) \left(p_\lambda(t, v, \boldsymbol{w}(\boldsymbol{z}))\, p_\mu(t, v, \boldsymbol{w}(\boldsymbol{z})) - \delta_{\lambda,\mu}\, p_\lambda(t, v, \boldsymbol{w}(\boldsymbol{z}))\right),$$

for $\lambda, \mu = 1, \cdots, K-1$. Thus, the Hessian matrix $(\partial^2 Q_1(\boldsymbol{z}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})/\partial z_\lambda \partial z_\mu)$ is negative semi-definite since

$$\begin{aligned}
&\sum_{\lambda,\mu=1}^{K-1} \frac{\partial^2 Q_1(\boldsymbol{z}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})}{\partial z_\lambda\, \partial z_\mu}\, x_\lambda x_\mu \\
&= \sum_{(v,t,k)\in\mathcal{D}_T} \sum_{j=1}^{K} q_{v,t,k,j}(\overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}) \left(\left(\sum_{\lambda=1}^{K-1} p_\lambda(t, v, \boldsymbol{w}(\boldsymbol{z}))\, x_\lambda\right)^2 - \sum_{\lambda=1}^{K-1} p_\lambda(t, v, \boldsymbol{w}(\boldsymbol{z}))\, {x_\lambda}^2\right) \\
&= -\sum_{(v,t,k)\in\mathcal{D}_T} \sum_{j=1}^{K} q_{v,t,k,j}(\overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}}) \left(\sum_{\lambda=1}^{K-1} p_\lambda(t, v, \boldsymbol{w}(\boldsymbol{z}))\left(x_\lambda - \sum_{\mu=1}^{K-1} p_\mu(t, v, \boldsymbol{w}(\boldsymbol{z}))\, x_\mu\right)^2\right. \\
&\qquad\qquad \left. + \left(1 - \sum_{\lambda=1}^{K-1} p_\lambda(t, v, \boldsymbol{w}(\boldsymbol{z}))\right)\left(\sum_{\mu=1}^{K-1} p_\mu(t, v, \boldsymbol{w}(\boldsymbol{z}))\, x_\mu\right)^2\right) \\
&\leq 0,
\end{aligned}$$

for any $(x_1, \cdots, x_{K-1}) \in \mathbf{R}^{K-1}$. Hence, by solving the equations $\partial Q_1(\boldsymbol{z}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})/\partial z_\lambda = 0$, $(\lambda = 1, \cdots, K-1)$ (see Eq. (17)), we can find the value of $\boldsymbol{z}$ that maximizes $\mathcal{Q}_1(\boldsymbol{z}; \overline{\boldsymbol{z}}, \overline{\boldsymbol{\alpha}})$. We employed a standard Newton Method in our experiments.

Next, we estimate the value of $\boldsymbol{\alpha}$ that maximizes $Q_2(\boldsymbol{\alpha}; \overline{\mathbf{z}}, \overline{\boldsymbol{\alpha}})$. From Eqs. (11) and (15), we have

$$Q_2(\boldsymbol{\alpha}; \overline{\mathbf{z}}, \overline{\boldsymbol{\alpha}}) = \sum_{(v,t,k)\in\mathcal{D}_T} \left(q_{v,t,k,k}(\overline{\mathbf{z}}, \overline{\boldsymbol{\alpha}}) \log(1-\alpha_v) + (1 - q_{v,t,k,k}(\overline{\mathbf{z}}, \overline{\boldsymbol{\alpha}})) \log\left(\frac{\alpha_v}{K-1}\right)\right).$$

Note that $Q_2(\boldsymbol{\alpha}; \overline{\mathbf{z}}, \overline{\boldsymbol{\alpha}})$ is also a convex function of $\boldsymbol{\alpha}$. Therefore, we obtain the unique solution $\boldsymbol{\alpha}$ that maximizes $Q(\mathbf{z}, \boldsymbol{\alpha}; \overline{\mathbf{z}}, \overline{\boldsymbol{\alpha}})$ as follows:

$$\alpha_v = \frac{1}{|\mathcal{D}_T(v)|} \sum_{(t,k)\in\mathcal{D}_T(v)} (1 - q_{v,t,k,k}(\overline{\mathbf{z}}, \overline{\boldsymbol{\alpha}})),$$

for each $v \in V$, where $\mathcal{D}_T(v) = \{(t,k); (v,t,k) \in \mathcal{D}_T\}$.

5 Experimental Evaluation

Using large real networks, we experimentally investigate the performance of the proposed learning method. We show the results of the estimation error of anti-majoritarian tendency, and the accuracies of detecting nodes with high anti-majoritarian tendency (*i.e.*, anti-majority opinionists) and nodes with low anti-majoritarian tendency (*i.e.*, majority opinionists), respectively.

5.1 Experimental Settings

We used three datasets of large real networks, which are all bidirectional connected networks and exhibit many of the key features of social networks. The first one is a trackback network of Japanese blogs used by [10] and has 12,047 nodes and 79,920 directed links (the Blog network). The second one is a network derived from the Enron Email Dataset [13] by extracting the senders and the recipients and linking those that had bidirectional communications. It has 4,254 nodes and 44,314 directed links (the Enron network). The third one is a network of people derived from the "list of people" within Japanese Wikipedia, also used by [10] and has 9,481 nodes and 245,044 directed links (the Wikipedia network).

We drew the true anti-majoritarian tendency α_v^* of each node $v \in V$ from the beta distribution of $a = b = 2$, and set the true opinion values as follows:

$$w_k^* = 5 - \frac{4(k-1)}{K-1}, \quad (k = 1, \cdots, K).$$

Note that the average value of α_v is expected to be 0.5, *i.e.*, $\alpha = 0.5$, and

$$w_1^* = 5,\ w_2^* = 5 - \frac{4}{K-1},\ \cdots,\ w_K^* = 1.$$

For each of three networks, we selected the initial opinion of each node uniformly at random, and generated the opinion diffusion data $\mathcal{D}_T$ of time span $[0, T]$ based on the true VwMV model. Then, we investigated the problem of estimating the anti-majoritarian tendency from the observed data $\mathcal{D}_T$.

We measured the error in estimating the anti-majoritarian tendency by estimation error $\mathcal{E}$,

$$\mathcal{E} = \frac{1}{|V|} \sum_{v \in V} |\hat{\alpha}_v - \alpha_v^*|,$$

where each $\hat{\alpha}_v$ denotes the estimated anti-majoritarian tendency of node v. We also measured the accuracies of detecting the high and the low anti-majoritarian tendency nodes by F-measures $\mathcal{F}_A$ and $\mathcal{F}_N$, respectively. Here, $\mathcal{F}_A$ and $\mathcal{F}_N$ are defined as follows:

$$\mathcal{F}_A = \frac{2|\hat{A} \cap A^*|}{|\hat{A}| + |A^*|}, \quad \mathcal{F}_N = \frac{2|\hat{N} \cap N^*|}{|\hat{N}| + |N^*|},$$

where A^* and $\hat{A}$ are the sets of the true and the estimated top 15% nodes of high anti-majoritarian tendency, respectively, and N^* and $\hat{N}$ are the sets of the true and the estimated top 15% nodes of low anti-majoritarian tendency, respectively.

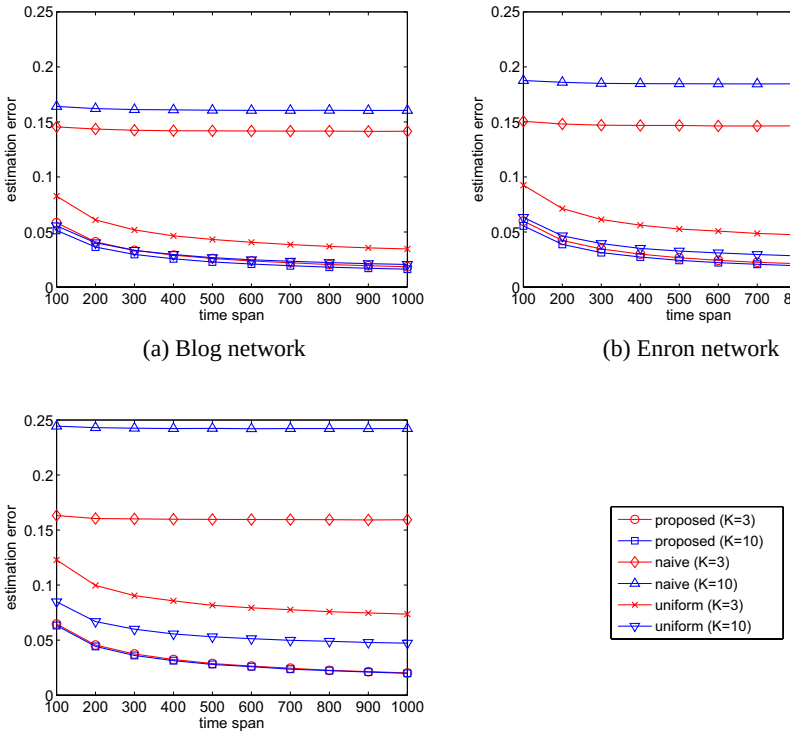
5.2 Comparison Methods

In order to investigate the importance of introducing the opinion values, we first compared the proposed method with the same VwMV model in which the opinion values are constrained to take a uniform value and the anti-majoritarian tendency of each node is the only parameter to be estimated. We refer to the method as the *uniform value method*. We also compared the proposed method with the naive approach in which the anti-majoritarian tendency of a node is estimated by simply counting the number of opinion updates in which the opinion chosen by the node is the minority's opinion in its neighborhood. We refer to the method as the *naive method*.

5.3 Experimental Results

We examined the results for both a small ($K = 3$) and a large ($K = 10$) K. Figures 2a, 2b and 2c show the estimation error $\mathcal{E}$ of each method as a function of time span T. Figures 3a, 3b and 3c show the F-measure $\mathcal{F}_A$ of each method as a function of time span T. Figures 4a, 4b and 4c show the F-measure $\mathcal{F}_N$ of each method as a function of time span T. Here, we repeated the same experiment five times independently, and plotted the average over the five results.

As expected, $\mathcal{E}$ decreases, and $\mathcal{F}_A$ and $\mathcal{F}_N$ increase as T increases (*i.e.*, the amount of training data $\mathcal{D}_T$ increases). We observe that the proposed method performs the best, the uniform value method follows, and the naive method behaves very poorly for all the networks. The proposed method can detect both the anti-majority and the majority opinionists with the accuracy greater than 90% at $T = 1000$ for all cases. We can also see that the proposed method is not sensitive to both K and the network structure, but the other two methods are so. For example, although the uniform value method of $K = 10$ performs well in $\mathcal{F}_A$ for the Blog and Enron networks, it does not so in $\mathcal{F}_A$ for the Wikipedia network, and in $\mathcal{F}_N$ for all the networks. Moreover, the uniform value method of $K = 3$ does not work well for all the cases. These results clearly demonstrate the advantage of the proposed method, and it does not seem feasible to detect even roughly



(a) Blog network

(b) Enron network

(c) Wikipedia network

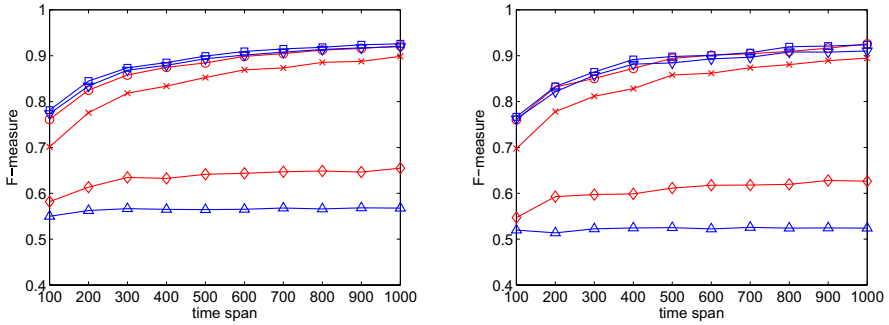
Fig. 2: Results for estimation errors of anti-majoritarian tendency.

the high anti-majoritarian tendency nodes and the low anti-majoritarian tendency nodes without using the explicit model and solving the optimization problem.

Here, we also note that the proposed method accurately estimated the opinion values. In fact, the average estimation errors of opinion value were less than 1% at $T = 1000$ for all cases. Moreover, we note that the processing times of the proposed method at $T = 1000$ for $K = 3$ and $K = 10$ were less than 3 min. and 4 min., respectively. All our experiments were undertaken on a single PC with an Intel Core 2 Duo 3GHz processor, with 2GB of memory, running under Linux.

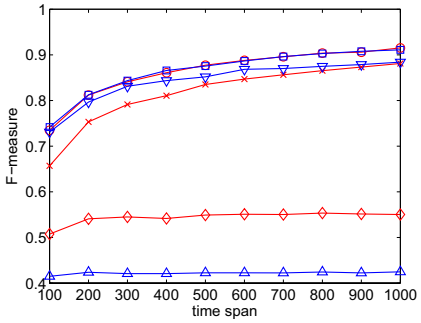
6 Conclusion

We addressed the problem of how different opinions with different values spread over a social network under the presence of anti-majority opinionists by Value-weighted Mixture Voter Model which combines the value-weighted voter and the anti-voter models both with multiple opinions. The degree of anti-majority (anti-majoritarian tendency) is quantified by the weight of the two models, and is treated as a parameter. We formulated the model in the machine learning framework, and learned the anti-majoritarian



(a) Blog network

(b) Enron network



(c) Wikipedia network

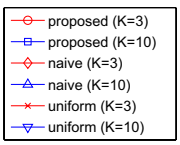
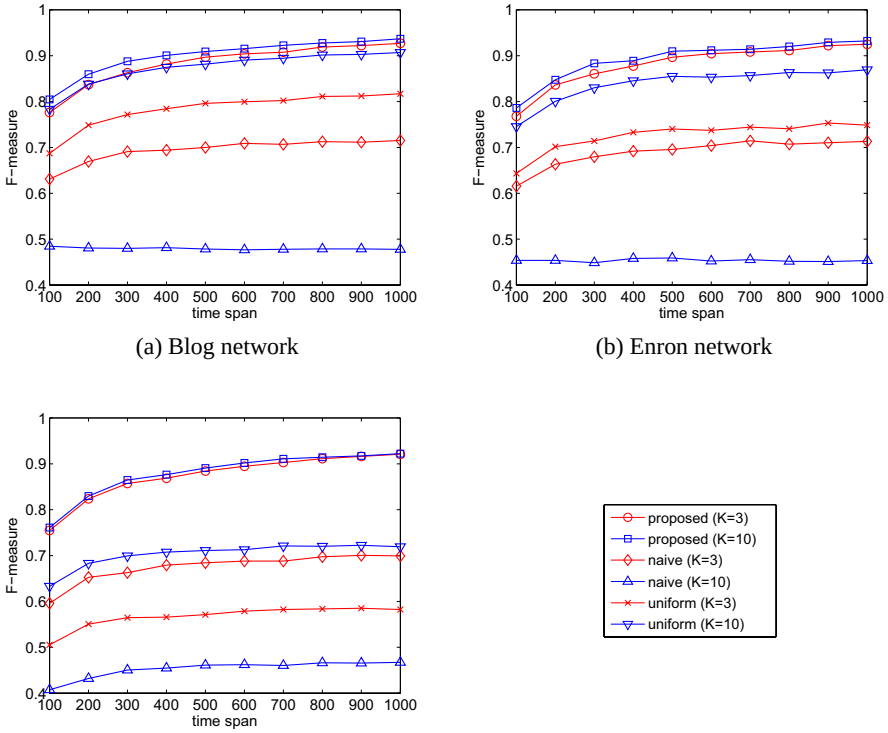


Fig. 3: Results for accuracies of extracting nodes with high anti-majoritarian tendency.

tendency of each node and the value of each opinion from a sequence of observed opinion diffusion such that the likelihood of the model's generating the data is maximized.

The iterative parameter update algorithm is efficient and correctly identifies both the anti-majoritarian tendency and the opinion value if there are enough data. We confirmed this by applying the algorithm to three real world social networks (Blog, Enron and Wikipedia) under various situations. We compared the results with the naive approach in which the anti-majoritarian tendency is estimated by simply counting the number of opinion updates such that the chosen opinion is the same as the minority's opinion. The naive approach behaves very poorly and our algorithm far outperformed it.

The opinion share crucially depends on the anti-majoritarian tendency and it is important to be able to accurately estimate it. The model learned by the proposed algorithm can be used to predict future opinion share and provides a useful tool to do various analyses. The theoretical analysis showed that in a situation where the local opinion share can be approximated by the average opinion share over the whole network, the opinion with the highest value does not necessarily prevails when the values are non-uniform, which is in contrast to the result of the value-weighted voter model (winner-take-all), whereas the opinion share prediction problem becomes ill-defined when the opinion



(a) Blog network

(b) Enron network

(c) Wikipedia network

Fig. 4: Results for accuracies of extracting nodes with low anti-majoritarian tendency.

values are uniform, *i.e.*, any opinion can win, which is the same as in the value-weighted voter model. The simulation results support that this holds for typical real world social networks. Our immediate future work is to apply the model to an interesting problem of influential maximization for opinion diffusion under the presence of anti-majority opinionists.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 23500194).

References

1. Castellano, C., Munoz, M.A., Pastor-Satorras, R.: Nonlinear *q*-voter model. Physical Review E 80, Article 041129 (2009)

2. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09). pp. 199–208 (2009)
3. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: Proceedings of the 10th IEEE International Conference on Data Mining (ICDM'10). pp. 88–97 (2010)
4. Crandall, D., Cosley, D., Huttenlocner, D., Kleinberg, J., Suri, S.: Feedback effects between similarity and sociai infiuence in online communities. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08). pp. 160–168 (2008)
5. Domingos, P.: Mining social networks for viral marketing. IEEE Intelligent Systems 20, 80–82 (2005)
6. Even-Dar, E., Shapira, A.: A note on maximizing the spread of influence in social networks. In: Proceedings of the 3rd International Workshop on Internet and Network Economics (WINE'07). pp. 281–286 (2007)
7. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. SIGKDD Explorations 6, 43–52 (2004)
8. Holme, P., Newman, M.E.J.: Nonequilibrium phase transition in the coevolution of networks and opinions. Physical Review E 74, Article 056108 (2006)
9. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03). pp. 137–146 (2003)
10. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data 3, Article 9 (2009)
11. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Extracting influential nodes on a social network for information diffusion. Data Mining and Knowledge Discovery 20, 70–97 (2010)
12. Kimura, M., Saito, K., Ohara, K., Motoda, H.: Learning to predict opinion share in social networks. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10). pp. 1364–1370 (2010)
13. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: Proceedings of the 15th European Conference on Machine Learning (ECML'04). pp. 217–226 (2004)
14. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. ACM Transactions on the Web 1, Article 5 (2007)
15. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07). pp. 420–429 (2007)
16. Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45, 167–256 (2003)
17. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Physical Review E 66, Article 035101 (2002)
18. Röllin, A.: Translated poisson approximation using exchangeable pair couplings. Annals of Applied Probablity 17, 1596–1614 (2007)
19. Sood, V., Redner, S.: Voter model on heterogeneous graphs. Physical Review Letters 94, Article 178701 (2005)
20. Wu, F., Huberman, B.A.: How public opinion forms. In: Proceedings of the 4th International Workshop on Internet and Network Economics (WINE'08). pp. 334–341 (2008)
21. Yang, H., Wu, Z., Zhou, C., Zhou, T., Wang, B.: Effects of social diversity on the emergence of global consensus in opinion dynamics. Physical Review E 80, Article 046108 (2009)

Efficient Detection of Hot Span in Information Diffusion from Observation

Kouzou Ohara[1] **Kazumi Saito**[2] **Masahiro Kimura**[3] **Hiroshi Motoda**[4]
[1]College of Science and Engineering, Aoyama Gakuin University, ohara@it.aoyama.ac.jp
[2]School of Administration and Informatics, University of Shizuoka, k-saito@u-shizuoka-ken.ac.jp
[3]Department of Electronics and Informatics, Ryukoku University, kimura@rins.ryukoku.ac.jp
[4]Institute of Scientific and Industrial Research, Osaka University, motoda@ar.sanken.osaka-u.ac.jp

Abstract

We addressed the problem of detecting the change in behavior of information diffusion from a small amount of observation data, where the behavior changes were assumed to be effectively reflected in changes in the diffusion parameter value. The problem is to detect where in time and how long this change persisted and how big this change is. We solved this problem by searching the change pattern that maximizes the likelihood of generating the observed diffusion sequences. The naive learning algorithm has to iteratively update the patten boundaries, each requiring optimization of diffusion parameters by the EM algorithm, and is very inefficient. We devised a very efficient search algorithm using the derivative of likelihood which avoids parameter value optimization during the search. The results tested using three real world network structures confirmed that the algorithm can efficiently identify the correct change pattern. We further compared our algorithm with the naive method that finds the best combination of change boundaries by an exhaustive search through a set of randomly selected boundary candidates, and showed that the proposed algorithm far outperforms the native method both in terms of accuracy and computation time.

1 Introduction

Social networking is now an important part of our daily lives, and our behavioral patterns are substantially affected by the communication through these networks [Newman *et al.*, 2002; Newman, 2003; Gruhl *et al.*, 2004; Domingos, 2005; Leskovec *et al.*, 2006]. It has been shown that a social network has many interesting properties, e.g. power law for node degree distribution, large clustering coefficient, positive degree correlation, etc. [Wasserman and Faust, 1994], which affect how the information actually diffuses through the network, and researchers have devised several important measures to characterize these features based on the topology/structure of the network [Wasserman and Faust, 1994; Bonacichi, 1987; Katz, 1953]. These measures, called centrality measures, are expected to be used to identify important nodes in the network. However, recent studies have shown that it is important to consider the diffusion mechanism explicitly and the measures based on network structure alone are not enough to identify the important nodes [Kimura *et al.*, 2009; 2010a].

Information diffusion is modeled typically by a probabilistic model. Most representative and fundamental ones are independent cascade (IC) model [Goldenberg *et al.*, 2001; Kempe *et al.*, 2003], linear threshold (LT) model [Watts, 2002; Watts and Dodds, 2007] and their extensions that include incorporating asynchronous time delay [Saito *et al.*, 2009]. Explicit use of these models to solve such problems as the *influence maximization problem* [Kempe *et al.*, 2003; Kimura *et al.*, 2010a] and the *contamination minimization problem* [Kimura *et al.*, 2009] clearly shows the advantage of the model. The identified influential nodes and links are considerably different from the ones identified by the centrality measures. However, use of these models brings in yet another difficulty. They have parameters that need be specified in advance, e.g. diffusion probabilities for the IC model, and weights for the LT model, and their true values are not known in practice. A series of studies by [Saito *et al.*, 2009; 2010] have shown one way of solving this problem in which they used a limited amount of observed information diffusion data and trained/learned the model such that the likelihood of generating the observed data by the model is maximized.

This paper is in the same line of these studies, but addresses a different aspect of information diffusion. Almost all of the work so far assumed that the model is stationary. We note that our behavior is affected not only by the behaviour of our neighbors but also by other external factors. The model only accounts for the interaction with neighbors. The problem we address here is to detect the change of the model from a limited amount of observed information diffusion data. If this is possible, we can infer that something unusual happened during a particular period of time by simply analyzing the limited amount of data.

This is in some sense the same, in the spirit, with the work by [Kleinberg, 2002] and [Swan and Allan, 2000]. They noted a huge volume of the data stream, tried to organize it and extract structures behind it. This is done in a retrospective framework, i.e. assuming that there is a flood of abundant data already and there is a strong need to understand it. Our aim is not exactly the same as theirs. We are interested in de-

tecting changes which is hidden in the data. We also follow the same retrospective approach, i.e. we are not predicting the future, but we are trying to understand the phenomena that happened in the past. There are many factors that bring in changes and the model cannot accommodate all of them. We formalize this as the unknown changes in the diffusion parameter value, and we reduce the problem to that of detecting where in time and how long this change persisted and how big this change is. We call the period where the parameter takes anomalous values as "hot span" and the rest as "normal span". To make the analysis simple, we limit the diffusion model to the asynchronous independent cascade model (AsIC) [Saito *et al.*, 2009] and the form of change to a rect-linear one, that is, the diffusion parameter changes to a new large value, persists for a certain period of time and is restored to the original value and stays the same thereafter [1]. In this simplified setting, detecting the hot span is equivalent to identifying the time window where the parameter value is high and estimating the parameter values both in hot and normal spans.

To this end, we use the same parameter optimization algorithm as in [Saito *et al.*, 2009], i.e. the EM algorithm that iteratively updates the values to maximize the model's likelihood of generating the observed data sequences. The problem here is more difficult because it has another loop to search for the hot span on top of the above loop. The naive learning algorithm has to iteratively update the patten boundaries requiring the parameter value optimization for each combination, which is a very inefficient procedure. Our main contribution is that we devised a very efficient general search algorithm which avoids the inner loop optimization by using the information of the first order derivative of the likelihood with respect to the diffusion parameters. We tested its performance using the structures of three real world networks (blog, Coauthorship and Wikipedia), and confirmed that the algorithm can efficiently identify the hot span correctly as well as the diffusion parameter values. We further compared our algorithm with the naive method that finds the best combination of change boundaries by an exhaustive search from a set of randomly selected boundary candidates, and showed that the proposed algorithm far outperforms the native method both in terms of accuracy and computation time.

2 Information Diffusion Model

The AsIC model we use in this paper incorporates asynchronous time delay into the independent cascade (IC) model which does not account for time-delay, reflecting that each node changes its state asynchronously in reality. We recall the definition of the AsIC model below, in which we consider choosing a delay-time from the exponential distribution for the sake of convenience, but of course other distributions such as power-law and Weibull can be employed.

Let $G = (V, E)$ be a directed graph, where V and E ($\subset V \times V$) are the sets of all the nodes and the links. For any $v \in V$, the set of all the nodes that have links from v is denoted by $F(v) = \{u \in V;\ (v, u) \in E\}$ and the set of all the nodes that have links to v by $B(v) = \{u \in V;\ (u, v) \in E\}$. Each node has one of the two states (active and inactive), and the nodes are called *active* if they have been influenced. It is assumed that nodes can switch their states only from inactive to active.

The AsIC model has two types of parameters $p_{u,v}$ and $r_{u,v}$ with $0 < p_{u,v} < 1$ and $r_{u,v} > 0$, where $p_{u,v}$ and $r_{u,v}$ are referred to as the diffusion probability through link (u, v) and the time-delay parameter through link (u, v), respectively. The information diffusion process unfolds in continuous-time t, and proceeds from a given initial active node in the following way. When a node u becomes active at time t, it is given a single chance to activate each currently inactive node $v \in F(u)$. A delay-time δ is chosen from the exponential distribution with parameter $r_{u,v}$. The node u attempts to activate the node v if v has not been activated by time $t + \delta$, and succeeds with probability $p_{u,v}$. If u succeed, v will become active at time $t+\delta$. The information diffusion process terminates if no more activations are possible.

3 Problem Setting

We address the *hot span detection problem*. In this problem, we assume that some change has happened in the way the information diffuses, and we observe the diffusion sequences of a certain topic in which the change is embedded, and consider detecting where in time and how long this change persisted and how big this change is. We place a constraint that $p_{u,v}$ and $r_{u,v}$ do not depend on link (u, v), i.e. $p_{u,v} = p$, $r_{u,v} = r$ $(\forall (u, v) \in E)$, which should be acceptable noting that we can naturally assume that people behave quite similarly when talking about the same topic (see Section 6).

Let $[T_1, T_2]$ denote the hot span of the diffusion of a topic, and let p_1 and p_2 denote the values of the diffusion probability of the AsIC model for the normal span and the hot span, respectively. Note that $p_1 < p_2$. A diffusion result of the topic is represented as a set of pairs of active nodes and their activation times; i.e. $\{(u, t_u), (v, t_v), \cdots\}$. We consider a diffusion result D that is generated by the AsIC model with p_1 for the period $[0, T_1)$, p_2 for the period $[T_1, T_2]$ and p_1 for the period (T_2, ∞), where the time-delay parameter does not change and takes the same value r for the entire period $[0, \infty)$. We refer to the set D as a *diffusion result with a hot span*. The problem is reduced to detecting $[T_1, T_2]$ and estimating p_1 and p_2 from the observed diffusion results. Extensions of this problem setting is discussed later (see Section6).

Figure 1 shows examples of diffusion samples with a hot span based on the AsIC model, where the parameters are set at $p_1 = 0.1$, $p_2 = 0.3$, $r = 1.0$, $T_1 = 10$ and $T_2 = 20$. The network used is the blog network described later in Subsection 5.1. We plotted the ratio of active nodes (the number of active nodes at a time step t divided by the number of total active nodes over the whole time span) for five independent simulations, each from a randomly chosen initial source node at time $t = 0$. We can clearly see bursty activities around the hot span $[T_1 = 10, T_2 = 20]$. However, each curve behaves differently, i.e., some has its bursty activities only in the first half, some other has them only in the last half, and yet some other has two peaks during the hot span. This means that it is quite difficult to accurately detect the true hot span from

[1] We discuss that the basic algorithm can be extended to more general change patterns in Section 6, and shows that it works for two distinct rect-linear patterns.

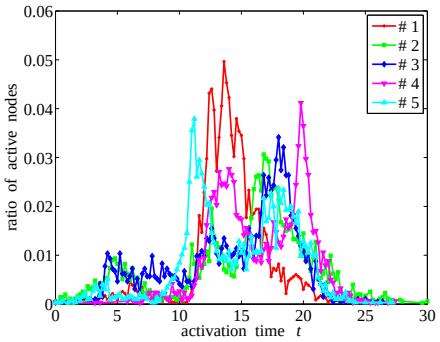


Figure 1: Information diffusion in the blog network with a hot span for the AsIC model.

only a single diffusion sample. Methods that use only the observed bursty activities, including those proposed by [Swan and Allan, 2000] and [Kleinberg, 2002] would not work. We believe that an explicit use of underlying diffusion model is essential to solve this problem. It is crucially important to detect the hot span precisely in order to identify the external factors which caused the behavioral changes.

4 Hot Span Detection Methods

Let $\{D_m;\ m = 1, \cdots, M\}$ be a set of M independent information diffusion results, where $D_m = \{(u, t_{m,u}),\ (v, t_{m,v}),\ \cdots\}$. Each D_m is associated with the observed initial time $\phi_m = \min\{t_{m,v};\ (v, t_{m,v}) \in D_m\}$, and the observed final time $\Phi_m \geq \max\{t_{m,v};\ (v, t_{m,v}) \in D_m\}$. We express our observation data by $\mathcal{D}_M = \{(D_m, \Phi_m);\ m = 1, \cdots, M\}$. For any $t \in [\phi_m, \Phi_m]$, we set $C_m(t) = \{v;\ (v, t_{m,v}) \in D_m,\ t_{m,v} < t\}$. Namely, $C_m(t)$ is the set of active nodes before time t in the mth diffusion result. For convenience sake, we use C_m as referring to the set of all the active nodes in the mth diffusion result.

4.1 Parameter Learning Framework

The following logarithmic likelihood function $\mathcal{L}(\mathcal{D}_M; p, r)$ has been derived to estimate the values of p and r from $\mathcal{D}_M$ for the AsIC model in case there is no hot span [Saito *et al.*, 2009],

$$\begin{aligned}\mathcal{L}(\mathcal{D}_M; p, r) &= \sum_{m=1}^{M} \mathcal{L}((D_m, \Phi_m); p, r) \\ &= \sum_{m=1}^{M} \sum_{v \in C_m} \left(\log h_{m,v} + \sum_{w \in F(v) \setminus C_m} \log g_{m,v,w} \right), \quad (1)\end{aligned}$$

where $h_{m,v}$ is the probability density that a node $v \in D_m$ with $t_{m,v} > 0$ is activated at a time $t_{m,v}$, and $g_{m,v,w}$ is the probability that a node w is not activated by a node v within $[\phi_m, \Phi_m]$, where there exists a link $(v, w) \in E$ and $v \in C_m$. The values of p and r can be stably obtained by maximizing Eq. (1) using the EM algorithm [Saito *et al.*, 2009].

The following parameter switching applies for a hot span $S = [T_1, T_2]$ where $\mathcal{N}_m$ and $\mathcal{H}_m$ denote the sets of active nodes in the m-th diffusion result during the normal and the hot spans, respectively.

$$p = \begin{cases} p_1 & \text{if } v \in \mathcal{N}_m(S),\ \mathcal{N}_m(S) = C_m(T_1) \cup (C_m \setminus C_m(T_2)), \\ p_2 & \text{if } v \in \mathcal{H}_m(S),\ \mathcal{H}_m(S) = C_m(T_2) \setminus C_m(T_1). \end{cases}$$

Then, an extended objective function $\mathcal{L}(\mathcal{D}_M; p_1, p_2, r, S)$ can be defined by adequately modifying Eq. (1) under this switching scheme. Clearly, $\mathcal{L}(\mathcal{D}_M; p_1, p_2, r, S)$ is expected to be maximized by setting S to the true span $S^* = [T_1^*, T_2^*]$ if a substantial amount of data $\mathcal{D}_M$ is available. Thus, our problem is to find the following $\hat{S}$.

$$\hat{S} = \arg\max_S \mathcal{L}(\mathcal{D}_M; \hat{p}_1, \hat{p}_2, \hat{r}, S), \quad (2)$$

where $\hat{p}_1$, $\hat{p}_2$, and $\hat{r}$ denote the maximum likelihood estimators for a given S.

In order to obtain $\hat{S}$, we need to prepare a reasonable set of candidate spans, denoted by $\mathcal{S}$. One way of doing so is to construct $\mathcal{S}$ by considering all pairs of observed activation time points: $\mathcal{S} = \{S = [t_1, t_2] : t_1 < t_2, t_1 \in \mathcal{T}, t_2 \in \mathcal{T}\}$, where $\mathcal{T} = \{t_1, \cdots, t_N\}$ is a set of activation time points in $\mathcal{D}_M$.

4.2 Naive Method

Now we describe the naive method, which has two iterative loops. In the inner loop we first obtain the maximum likelihood estimators, $\hat{p}_1$, $\hat{p}_2$, and $\hat{r}$, for each candidate S by maximizing $\mathcal{L}(\mathcal{D}_M; p_1, p_2, r, S)$ using the EM algorithm. In the outer loop we select the optimal $\hat{S}$ which gives the largest $\mathcal{L}(\mathcal{D}_M; \hat{p}_1, \hat{p}_2, \hat{r}, S)$ value. However, this can be extremely inefficient when N is large. To make it work with a reasonable computational cost, we restrict the number of candidate time points N to a smaller value K by selecting K points from $\mathcal{T}$, i.e., we construct $\mathcal{S}_K = \{S = [t_1, t_2] : t_1 < t_2, t_1 \in \mathcal{T}_k, t_2 \in \mathcal{T}_K\}$, where $\mathcal{T}_K = \{t_1, \cdots, t_K\}$. Note that $|\mathcal{S}_K| = K(K-1)/2$, which is large when K is large.

4.3 Proposed Method

The naive method should be able to detect the hot span with a reasonable accuracy when K is set large at the expense of the computational cost, but the accuracy becomes poorer when K is set smaller to reduce the computational load. We propose a novel detection method which alleviates this problem and can efficiently and stably detect a hot span from $\mathcal{D}_M$.

We first obtain $\hat{p}$, and $\hat{r}$, based on the original objective function of Eq. (1), and focus on its first-order derivative with respect to p for each node at each individual activation time. Let $p_{u,v}$ be the diffusion parameter from a node u to a node v. The following formula holds for the maximum likelihood estimators due to the uniform parameter setting of Eq. (1) and the locally optimal condition.

$$\frac{\partial \mathcal{L}(\mathcal{D}_M; \hat{p}, \hat{r})}{\partial p} = \sum_{(u,v) \in E} \frac{\partial \mathcal{L}(\mathcal{D}_M; \hat{p}, \hat{r})}{\partial p_{u,v}} = 0. \quad (3)$$

Consider the following partial sum for a given $S = [T_1, T_2]$.

$$\mathcal{G}(S) = \sum_{m=1}^{M} \sum_{(u,v) \in E, u \in \mathcal{H}_m(S)} \frac{\partial \mathcal{L}((D_m, \Phi_m); \hat{p}, \hat{r})}{\partial p_{u,v}}. \quad (4)$$

Clearly, $\mathcal{G}(S)$ should be sufficiently large if $S \approx S^*$ due to our problem setting, which leads to $p_2 > \hat{p} > p_1$. Thus, the hot span S^* can be estimated by searching for $\hat{S}$ that maximizes $\mathcal{G}(S)$.

$$\hat{S} = \arg\max_{S \in \mathcal{S}} \mathcal{G}(S). \quad (5)$$

The nice thing here is that we can incrementally calculate $\mathcal{G}(S)$ by Eq. (6), where $\mathcal{T} = \{t_1, \cdots, t_N\}$ and $t_i < t_j$ if $i < j$.

$$\mathcal{G}([t_i, t_{j+1}]) = \mathcal{G}([t_i, t_j]) + \sum_{m=1}^{M} \sum_{\substack{(u,v)\in E \\ u\in C_m(t_{j+1})\setminus C_m(t_j)}} \frac{\partial \mathcal{L}((D_m, \Phi_m); \hat{p}, \hat{r})}{\partial p_{u,v}}. \quad (6)$$

The computational cost for examining each candidate span is much smaller than the naive method described above. Thus, we can use all the pairs to construct $\mathcal{S}$. We summarize our proposed method below.

1. Maximize $\mathcal{L}(\mathcal{D}_M; p, r)$ by using the EM algorithm.

2. Construct $\mathcal{T}$ and $\mathcal{S}$.

3. Detect $\hat{S}$ by Eq. (5) and output $\hat{S}$.

4. Maximize $\mathcal{L}(\mathcal{D}_M; p_1, p_2, r, \hat{S})$ by using the EM algorithm, and output $\hat{p}_1$, $\hat{p}_2$, and $\hat{r}$.

Here note that the proposed method requires maximization by using the EM algorithm only twice.

5 Experiments

We experimentally investigated how accurately the proposed method can estimate both the hot span and the diffusion probabilities in the hot and normal spans, as well as its efficiency, by comparing it with the naive method using three real world networks. We used three different values for K, *i.e.*, $K = 5$, 10, and 20 for the naive method.

The derivation assumed that there are multiple observed data sequences, but in the experiments we chose to learn from a single sequence, *i.e.*, $M = 1$, which is the most difficult situation.

5.1 Datasets

The three data are all bidirectionally connected networks. The first one is a trackback network of Japanese blogs used in [Kimura *et al.*, 2009], which has 12,047 nodes and 79,920 directed links (the blog network). The second one is a coauthorship network used in [Palla *et al.*, 2005], which has 12,357 nodes and 38,896 directed links (the Coauthorship network). The last one is a network of people that was derived from the "list of people" within Japanese Wikipedia, used in [Kimura *et al.*, 2009], and has 9,481 nodes and 245,044 directed links (the Wikipedia network).

For these networks, we generated diffusion samples with a hot span using the AsIC model. According to [Kempe *et al.*, 2003], we set the diffusion probability for the normal span, p_1, to be a value smaller than $1/d$, where d is the mean outdegree of a network, and set the diffusion probability for the hot span, p_2, to be three times larger than p_1. Thus, p_1 and p_2 are 0.1 and 0.3 for the blog network, 0.2 and 0.6 for the Coauthorship network, and 0.02 and 0.06 for the Wikipedia network, respectively. We fixed the time-delay parameter at 1 ($r = 1$) for all the networks because changing r works only for scaling the time axis of the diffusion results. We set the hot span to $[T_1 = 10, T_2 = 20]$ based on the observation on the preliminary experiments. In all we generated five information diffusion samples using these parameter values for each network, randomly selecting an initial active node for each diffusion sample.

5.2 Results

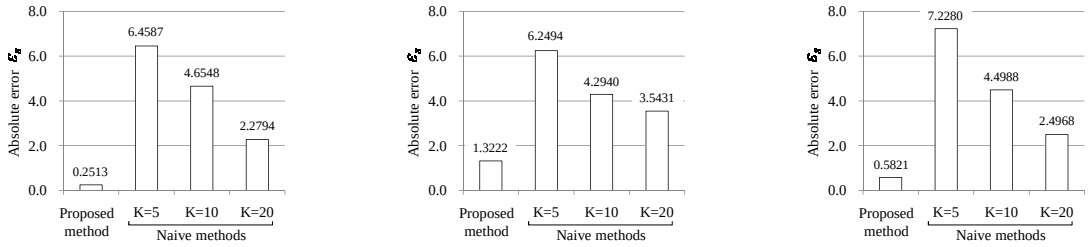
We compared the proposed method with the naive method in terms of 1) the accuracy of the estimated hot span $\hat{\mathcal{S}} = [\hat{T}_1, \hat{T}_2]$, 2) the accuracy of the diffusion probabilities p_1 (for the normal span) and p_2 (for the hot span), and 3) the computation time. Both the proposed and the naive methods were tested to each diffusion sample mentioned above, and the results were averaged over the five independent trials for each network.

Figure 2 shows the accuracy for $\hat{\mathcal{S}}$ in the absolute error $\mathcal{E}_s = |\hat{T}_1 - T_1| + |\hat{T}_2 - T_2|$. We see that the proposed method achieves a good accuracy, much better than the naive method for every network. As expected, $\mathcal{E}_s$ for the naive method decreases as K becomes larger. But, even in the best case ($K = 20$), its average error is about 3 to 10 times larger than that of the proposed method. Figure 3 shows the accuracy of p_1 and p_2 in the relative error $\mathcal{E}_p = |\hat{p}_1 - p_1|/p_1 + |\hat{p}_2 - p_2|/p_2$. Here again, the average relative error for the naive method decreases as K becomes larger. However, even in the best case ($K = 20$), it is about 2 to 3 times larger than that of the proposed method. We note that the average errors for the Coauthorship network are relatively large. This is because the number of active nodes within the normal span was relatively small for this network. Figure 4 shows the computation time. It is clear that the proposed method is much faster than the naive method. The significant difference is attributed to the difference in the number of runs of the EM algorithm. The proposed method executes the EM algorithm only twice: steps 1 and 4 in the algorithm (see Section 4.3). On the other hand, the naive method has to execute the EM algorithm once for every single candidate span $S \in \mathcal{S}_K$ which is $|\mathcal{S}_K| = K(K-1)/2$ times (see Section 4.2). Indeed, the computation time of the naive method for $K = 5$ is about 5 times larger for every network, which is consistent with $|\mathcal{S}_K| = 10$. This relation roughly holds also for the other two cases ($K = 10$ and $K = 20$). This means that even if the naive method could achieve a good accuracy by setting K to a sufficiently large value, it would require unacceptable computation time for such a large K.

In summary, we can say that the proposed method can detect and estimate the hot span and diffusion probabilities much more accurately and efficiently compared with the naive method. Here we mention that we could obtain much better results by using more than one diffusion sequence, say $M = 5$, but we have to omit the details due to space limitations.

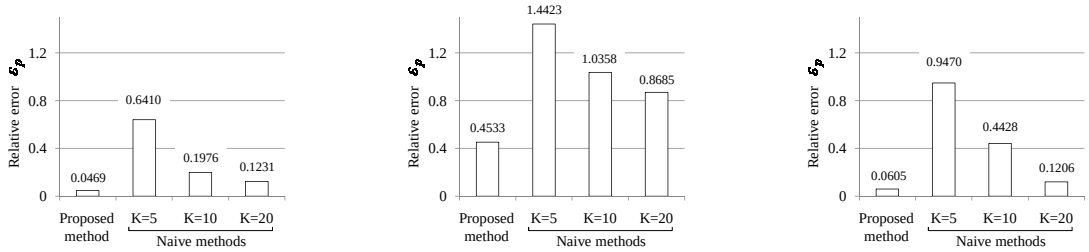
6 Discussion

We placed a simplifying constraint that the parameters $p_{u,v}$ and $r_{u,v}$ are link independent, i.e. $p_{u,v} = p$, $r_{u,v} = r$ ($\forall (u,v) \in E$), by focusing on single topic diffusion sequences. [Saito *et al.*, 2009; 2010] gave some evidences for this assumption. They examined 7,356 diffusion sequences for a real blogroll network containing 52,525 bloggers and 115,552 blogroll links, and experimentally confirmed that p and r that were learned from different diffusion sequences belonging to the same topic were quite similar for most of the topics. This



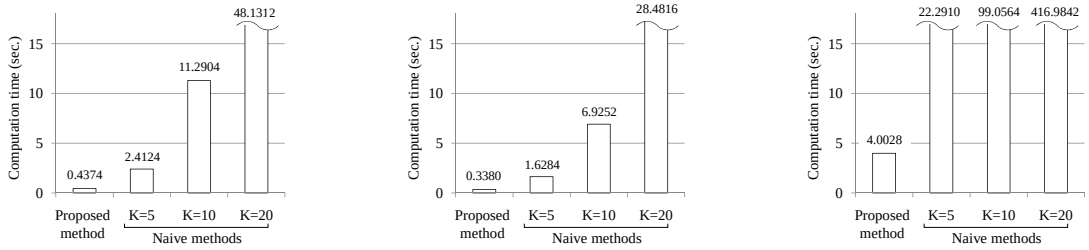
(a) Blog (b) Coauthorship (c) Wikipedia

Figure 2: Comparison in accuracies of the estimated hot span



(a) Blog (b) Coauthorship (c) Wikipedia

Figure 3: Comparison in accuracies of the estimated diffusion probability



(a) Blog (b) Coauthorship (c) Wikipedia

Figure 4: Comparison in computation time

observation naturally suggests that people behave quite similarly for the same topic.

In this paper, we considered AsIC model, but it is straightforward to apply the same technique to AsLT model [Saito *et al.*, 2010] and to their SIS versions in which each node is allowed to be activated multiple times. The same idea can naturally be applied to opinion formation model, e.g. value-weighted voter model [Kimura *et al.*, 2010b].

The change pattern considered here is the simplest one. We can assume a more intricate problem setting such that both p and r change for multiple distinct hot spans and the shape of change pattern p is not necessarily rect-linear. One possible extension is to approximate the pattern of any shape by J pairs of time interval each with its corresponding p_j, *i.e.*, $Z_J = \{([t_{j-1}, t_j], p_j);\ j = 1, \cdots J\}$ ($t_0 = 0, t_J = \infty$) and use a divide-and-conquer type greedy recursive partitioning, still employing the derivative of the likelihood function $\mathcal{G}$ as the main measure for search. More specifically, we first initialize $Z_1 = \{([0, \infty), \hat{p}_1)\}$ where $\hat{p}_1$ is the maximum likelihood estimator, and search for the first change time point t_1, which we expect to be the most distinguished one, by maximizing $|\mathcal{G}([t, \infty), \hat{p}_1)|$.[2] We recursively perform this operation J times by fixing the previously determined change points. When to stop can be determined by a statistical criterion such as AIC or MDL. This algorithm requires parameter optimization J times. Figure 5 is one of the preliminary results obtained for two distinct rect-linear patterns using five sequences ($M = 5$) in case of the blog network. MDL is used as the stopping criterion. The change pattern of p is almost perfectly detected with respect to both p_j and t_j ($J = 5$).

7 Conclusion

In this paper, we addressed the problem of detecting the change in behavior of information diffusion from a limited amount of observed diffusion sequences in a retrospective setting, assuming that the diffusion follows the asynchronous independent cascade (AsIC) model. We defined the "hot span" as the period during which the diffusion probability is changed to a relatively high value compared with the other periods (called the normal spans). A naive method to detect such a hot span would have to iteratively update the candidate hot span boundaries, each requiring parameter optimization such that the likelihood function is maximized. This is very inefficient and totally unacceptable. We developed a novel and general framework that avoids the inner loop optimization during search by making use of the first derivative of the likelihood function. It needs to optimize the pa-

[2] Note that the total sum of $\mathcal{G} = 0$.

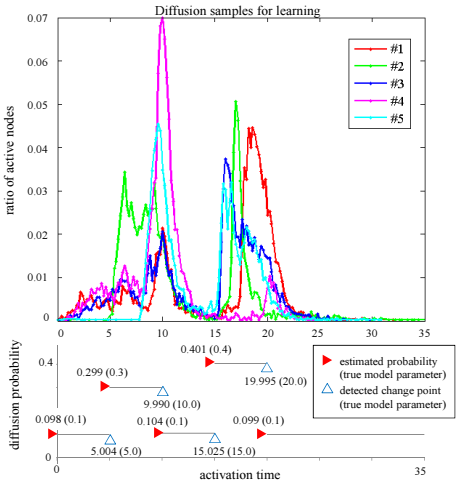


Figure 5: Information diffusion in the blog network with two hot spans for the AsIC model.

rameter values only twice by the iterative updating algorithm (EM algorithm), which reduces the computation times by 5 to 100 times, and is very efficient. We compared the proposed method with the naive method that considers only the randomly selected boundary candidates, by applying both the methods (the proposed and the naive) to information diffusion samples generated by simulation from three real world large networks, and confirmed that the proposed method far outperforms the naive method both in terms of accuracy and efficiency. Although we assumed a very simplified problem setting in this paper, the proposed method can be easily extended to solve more intricate problems. We showed one possible direction and the preliminary results obtained for two rect-linear shape hot spans was very promising. Our immediate future work is to evaluate our method using real world information diffusion samples with hot spans, as well as to deal with spatio-temporal hot span detection problems using more appropriate stochastic models under a similar problem solving framework.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 23500194).

References

[Bonacichi, 1987] P. Bonacichi. Power and centrality: A family of measures. *Amer. J. Sociol.*, 92:1170–1182, 1987.

[Domingos, 2005] P. Domingos. Mining social networks for viral marketing. *IEEE Intell. Syst.*, 20:80–82, 2005.

[Goldenberg *et al.*, 2001] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Market. Lett.*, 12:211–223, 2001.

[Gruhl *et al.*, 2004] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. *SIGKDD Expl.*, 6:43–52, 2004.

[Katz, 1953] L. Katz. A new status index derived from sociometric analysis. *Sociometry*, 18:39–43, 1953.

[Kempe *et al.*, 2003] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD 2003*, pages 137–146, 2003.

[Kimura *et al.*, 2009] M. Kimura, K. Saito, and H. Motoda. Blocking links to minimize contamination spread in a social network. *ACM Trans. Knowl. Discov. Data*, 3:9:1–9:23, 2009.

[Kimura *et al.*, 2010a] M. Kimura, K. Saito, R. Nakano, and H. Motoda. Extracting influential nodes on a social network for information diffusion. *Data Min. Knowl. Disc.*, 20:70–97, 2010.

[Kimura *et al.*, 2010b] M. Kimura, K. Saito, K. Ohara, and H. Motoda. Learning to predict opinion share in social networks. In *AAAI-10*, pages 1364–1370, 2010.

[Kleinberg, 2002] J. Kleinberg. Bursty and hierarchical structure in streams. In *KDD 2002*, pages 91–101, 2002.

[Leskovec *et al.*, 2006] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. In *EC'06*, pages 228–237, 2006.

[Newman *et al.*, 2002] M. E. J. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. *Phys. Rev. E*, 66:035101, 2002.

[Newman, 2003] M. E. J. Newman. The structure and function of complex networks. *SIAM Rev.*, 45:167–256, 2003.

[Palla *et al.*, 2005] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[Saito *et al.*, 2009] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Learning continuous-time information diffusion model for social behavioral data analysis. In *ACML 2009*, pages 322–337, 2009.

[Saito *et al.*, 2010] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Selecting information diffusion models over social networks for behavioral analysis. In *ECML PKDD 2010*, pages 180–195, 2010.

[Swan and Allan, 2000] R. Swan and J. Allan. Automatic generation of overview timelines. In *SIGIR 2000*, pages 49–56, 2000.

[Wasserman and Faust, 1994] S. Wasserman and K. Faust. *Social network analysis*. Cambridge Univ. Press, Cambridge, UK, 1994.

[Watts and Dodds, 2007] D. J. Watts and P. S. Dodds. Influence, networks, and public opinion formation. *J. Consum. Res.*, 34:441–458, 2007.

[Watts, 2002] D. J. Watts. A simple model of global cascades on random networks. *PNAS*, 99:5766–5771, 2002.

Learning Diffusion Probability based on Node Attributes in Social Networks

Kazumi Saito[1], Kouzou Ohara[2], Yuki Yamagishi[1], Masahiro Kimura[3], and Hiroshi Motoda[4]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp
[3] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. Information diffusion over a social network is analyzed by modeling the successive interactions of neighboring nodes as probabilistic processes of state changes. We address the problem of estimating parameters (diffusion probability and time-delay parameter) of the probabilistic model as a function of the node attributes from the observed diffusion data by formulating it as the maximum likelihood problem. We show that the parameters are obtained by an iterative updating algorithm which is efficient and is guaranteed to converge. We tested the performance of the learning algorithm on three real world networks assuming the attribute dependency, and confirmed that the dependency can be correctly learned. We further show that the influence degree of each node based on the link-dependent diffusion probabilities is substantially different from that obtained assuming a uniform diffusion probability which is approximated by the average of the link-dependent diffusion probabilities.

1 Introduction

The growth of Internet has enabled to form various kinds of large-scale social networks, through which a variety of information, e.g. news, ideas, hot topics, malicious rumors, etc.) spreads in the form of "word-of-mouth" communications, and it is noticeable to observe how much they affect our daily life style. The spread of information has been studied by many researchers [15, 14, 4, 1, 12, 7, 9]. The information diffusion models widely used are the *independent cascade (IC)* [2, 5, 7] and the *linear threshold (LT)* [21, 22] models. They have been used to solve such problems as the *influence maximization problem* [5, 8] and the *contamination minimization problem* [7, 20]. These two models focus on different information diffusion aspects. The IC model is sender-centered (push)

and each active node *independently* influences its inactive neighbors with given diffusion probabilities. The LT model is receiver-centered (pull) and a node is influenced by its active neighbors if their total weight exceeds the threshold for the node.

What is important to note is that both models have parameters that need be specified in advance: diffusion probabilities for the IC model, and weights for the LT model. However, their true values are not known in practice. This poses yet another problem of estimating them from a set of information diffusion results that are observed as time-sequences of influenced (activated) nodes. This falls in a well defined parameter estimation problem in machine learning framework. Given a generative model with some parameters and the observed data, it is possible to calculate the likelihood that the data are generated and the parameters can be estimated by maximizing the likelihood. To the best of our knowledge, we are the first to follow this line of research. We addressed this problem for the IC model [16] and devised the iterative parameter updating algorithm.

The problem with both the IC and LT models is that they treat the information propagation as a series of state changes of nodes and the changes are made in a synchronous way, which is equivalent to assuming a discrete time step. However, the actual propagation takes place in an asynchronous way along the continuous time axis, and the time stamps of the observed data are not equally spaced. Thus, there is a need to extend both models to make the state changes asynchronous. We have, thus, extended both the models to be able to simulate asynchronous time delay (the extended models are called AsIC and AsLT models) and showed that the same maximum likelihood approach works nicely [17–19] and recently extended the same approach to opinion propagation problem using the value-weighted voter model with multiple opinions [10]. There are other works which are close to ours that also attempted to solve the similar problem by maximizing the likelihood [3, 13], where the focus was on inferring the underlying network. In particular, [13] showed that the problem can effectively be transformed to a convex programming for which a global solution is guaranteed.

In this paper we also address the same problem using the AsIC model, but what is different from all of the above studies is that we try to learn the dependency of the diffusion probability and the time-delay parameter on the node attributes rather than learn it directly from the observed data. In reality the diffusion probability and the time-delay parameter of a link in the network must at least be a function of the attributes of the two connecting nodes, and ignoring this property does not reflect the reality. Another big advantage of explicitly using this relationship is that we can avoid overfitting problem. Since the number of links is much larger than the number of nodes even if the social network is known to be sparse, the number of parameters to learn is huge and we need prohibitively large amount of data to learn each individual diffusion probability separately. Because of this difficulty, many of the studies assumed that the parameter is uniform across different links or it depends only on the topic (not on the link that the topic passes through). Learning a function is much more realistic and does not require such a huge amount of data.

We show that the parameter updating algorithm is very efficient and is guaranteed to converge. We tested the performance of the algorithm on three real world networks assuming the attribute dependency of the parameters. The algorithm can correctly estimate both the diffusion probability and the time-delay parameter by way of node at-

tributes through a learned function, and we can resolve the deficiency of uniform parameter value assumption. We further show that the influence degree of each node based on the link-dependent diffusion probabilities (via learned function) is substantially different from that obtained assuming a uniform diffusion probability which is approximated by the average of the link-dependent diffusion probabilities, indicating that the uniform diffusion probability assumption is not justified if the true diffusion probability is link-dependent.

2 Diffusion Model

2.1 AsIC Model

To mathematically model the information diffusion in a social network, we first recall the AsIC model according to [19], and then extend it to be able to handle node attributes. Let $G = (V, E)$ be a directed network without self-links, where V and E ($\subset V \times V$) stand for the sets of all the nodes and links, respectively. For each node $v \in V$, let $F(v)$ be the set of all the nodes that have links from v, *i.e.*, $F(v) = \{u \in V;\ (v, u) \in E\}$, and $B(v)$ be the set of all the nodes that have links to v, *i.e.*, $B(v) = \{u \in V;\ (u, v) \in E\}$. We say a node is *active* if it has been influenced with the information; otherwise it is inactive. We assume that a node can switch its state only from inactive to active.

The AsIC model has two types of parameter $p_{u,v}$ and $r_{u,v}$ with $0 < p_{u,v} < 1$ and $r_{u,v} > 0$ for each link $(u, v) \in E$, where $p_{u,v}$ and $r_{u,v}$ are referred to as the diffusion probability and the time-delay parameter through link (u, v), respectively. Then, the information diffusion process unfolds in continuous-time t, and proceeds from a given initial active node in the following way. When a node u becomes active at time t, it is given a single chance to activate each currently inactive node $v \in F(u)$: u attempts to activate v if v has not been activated before time $t + \delta$, and succeeds with probability $p_{u,v}$, where δ is a delay-time chosen from the exponential distribution [1] with parameter $r_{u,v}$. The node v will become active at time $t + \delta$ if u succeed. The information diffusion process terminates if no more activations are possible.

2.2 Extension of AsIC Model for Using Node Attributes

In this paper, we extend the AsIC model to explicitly treat the attribute dependency of diffusion parameter through each link. Each node can have multiple attributes, each of which is either nominal or numerical. Let v_j be a value that node v takes for the j-th attribute, and J the total number of the attributes. For each link $(u, v) \in E$, we can consider the J-dimensional vector $\boldsymbol{x}_{u,v}$, each element of which is calculated by some function of u_j and v_j, *i.e.*, $x_{u,v,j} = f_j(u_j, v_j)$. Hereafter, for the sake of convenience, we consider the augmented $(J + 1)$-dimensional vector $\boldsymbol{x}_{u,v}$ by setting $x_{u,v,0} = 1$ as the link attributes. Then we propose to model both the diffusion probability $p_{u,v}$ and the

[1] We chose a delay-time from the exponential distribution in this paper for the sake of convenience, but other distributions such as power-law and Weibull can be employed.

time-delay parameter $r_{u,v}$ for each link $(u, v) \in E$ by the following formulae [2]:

$$p_{u,v} = p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{x}_{u,v})}, \quad r_{u,v} = r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi}) = \exp(\boldsymbol{\phi}^T \boldsymbol{x}_{u,v}), \tag{1}$$

where $\boldsymbol{\theta}^T = (\theta_0, \cdots, \theta_J)$ and $\boldsymbol{\phi}^T = (\phi_0, \cdots, \phi_J)$ are the $(J + 1)$-dimensional parameter vectors for diffusion probability and time-delay parameter, respectively. Note here that θ_0 and ϕ_0 correspond to the constant terms, and $\boldsymbol{\theta}^T$ stands for a transposed vector of $\boldsymbol{\theta}$.

Although our modeling framework does not depend on a specific form of function f_j, we limit the form to be the following: $x_{u,v,j} = \exp(-|u_j - v_j|)$ if the j-th node attribute is numerical; $x_{u,v,j} = \delta(u_j, v_j)$ if the j-th node attribute is nominal, where $\delta(u_j, v_j)$ is a delta function defined by $\delta(u_j, v_j) = 1$ if $u_j = v_j$; $\delta(u_j, v_j) = 0$ otherwise. Intuitively, the more similar u_j and v_j are, that is, the closer their attribute values are to each other, the larger the diffusion probability $p_{u,v}$ is if the corresponding parameter value θ_j is positive, and the smaller if it is negative. We can see the similar observation for the time-delay parameter $r_{u,v}$.

3 Learning Problem and Method

We consider an observed data set of M independent information diffusion results, $\mathcal{D}_M = \{D_m;\ m = 1, \cdots, M\}$. Here, each D_m represents a sequence of obserbation. It is given by a set of pairs of active node and its activation time, $D_m = \{(u, t_{m,u}),\ (v, t_{m,v}),\ \cdots\}$, and called the mth diffusion result. These sequences may partially overlap, *i.e.*, a node may appear in more than one sequence, but are treated separately according to the AsIC model. We denote by $t_{m,v}$ the activation time of node v for the mth diffusion result. Let T_m be the observed final time for the mth diffusion result. Then, for any $t \leq T_m$, we set $C_m(t) = \{v \in V;\ (v, t_{m,v}) \in D_m,\ t_{m,v} < t\}$. Namely, $C_m(t)$ is the set of active nodes before time t in the mth diffusion result. For convenience sake, we use C_m as referring to the set of all the active nodes in the mth diffusion result. For each node $v \in C_m$, we define the following subset of parent nodes, each of which had a chance to activate v, i.e., $\mathcal{B}_{m,v} = B(v) \cap C_m(t_{m,v})$.

3.1 Learning Problem

According to Saito et al. [17], we define the probability density $\mathcal{X}_{m,u,v}$ that a node $u \in \mathcal{B}_{m,v}$ activates the node v at time $t_{m,v}$, and the probability $\mathcal{Y}_{m,u,v}$ that the node v is not activated by a node $u \in \mathcal{B}_{m,v}$ within the time-period $[t_{m,u}, t_{m,v}]$.

$$\mathcal{X}_{m,u,v} = p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}) r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi}) \exp(-r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u})). \tag{2}$$

$$\mathcal{Y}_{m,u,v} = p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}) \exp(-r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u})) + (1 - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})). \tag{3}$$

Then, we can consider the following probability density $h_{m,v}$ that the node v is activated at time $t_{m,v}$:

[2] Note that both are smooth with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ and guarantee $0 < p < 1$ and $r > 0$.

$$h_{m,v} = \sum_{u \in \mathcal{B}_{m,v}} \mathcal{X}_{m,u,v} \left(\prod_{z \in \mathcal{B}_{m,v} \setminus \{u\}} \mathcal{Y}_{m,z,v} \right). = \prod_{z \in \mathcal{B}_{m,v}} \mathcal{Y}_{m,z,v} \sum_{u \in \mathcal{B}_{m,v}} \mathcal{X}_{m,u,v} (\mathcal{Y}_{m,u,v})^{-1}. \quad (4)$$

Next, we consider the following probability $g_{m,v,w}$ that the node w is not activated by the node v before the observed final time T_m.

$$g_{m,v,w} = p(\boldsymbol{x}_{v,w}, \boldsymbol{\theta}) \exp(-r(\boldsymbol{x}_{v,w}, \boldsymbol{\phi})(T_m - t_{m,v})) + (1 - p(\boldsymbol{x}_{v,w}, \boldsymbol{\theta})). \quad (5)$$

Here we can naturally assume that each information diffusion process finished sufficiently earlier than the observed final time, i.e., $T_m \gg \max\{t_{m,v}; (v, t_{m,v}) \in D_m\}$. Thus, as $T_m \to \infty$ in Equation (5), we can assume

$$g_{m,v,w} = 1 - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}). \quad (6)$$

By using Equations (4) and (6), and the independence properties, we can define the likelihood function $\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi})$ with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ by

$$\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi}) = \log \prod_{m=1}^{M} \prod_{v \in C_m} \left(h_{m,v} \prod_{w \in F(v) \setminus C_m} g_{m,v,w} \right). \quad (7)$$

In this paper, we focus on Equation (6) for simplicity, but we can easily modify our method to cope with the general one (i.e., Equation (5)). Thus, our problem is to obtain the values of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, which maximize Equation (7). For this estimation problem, we derive a method based on an iterative algorithm in order to stably obtain its solution.

3.2 Learning Method

Again, according to Saito et al. [17], we introduce the following variables to derive an EM like iterative algorithm.

$$\mu_{m,u,v} = \mathcal{X}_{m,u,v} (\mathcal{Y}_{m,u,v})^{-1} \Big/ \sum_{z \in \mathcal{B}_{m,v}} \mathcal{X}_{m,z,v} (\mathcal{Y}_{m,z,v})^{-1}.$$
$$\eta_{m,u,v} = p_{u,v} \exp(-r_{u,v}(t_{m,v} - t_{m,u})) \,/\, \mathcal{Y}_{m,u,v}.$$
$$\xi_{m,u,v} = \mu_{m,u,v} + (1 - \mu_{m,u,v}) \eta_{m,u,v}$$

Let $\bar{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\phi}}$ be the current estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively. Similarly, let $\bar{\mathcal{X}}_{m,u,v}$, $\bar{\mathcal{Y}}_{m,u,v}$, $\bar{\mu}_{m,u,v}$, $\bar{\eta}_{m,u,v}$, and $\bar{\xi}_{m,u,v}$ denote the values of $\mathcal{X}_{m,u,v}$, $\mathcal{Y}_{m,u,v}$, $\mu_{m,u,v}$, $\eta_{m,u,v}$, and $\xi_{m,u,v}$ calculated by using $\bar{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\phi}}$, respectively.

From Equations (4), (6) and (7), we can transform our objective function $\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi})$ as follows:

$$\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi}) = \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) - \mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}), \quad (8)$$

where $Q(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$ is defined by

$$Q(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) = Q_1(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) + Q_2(\boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$$

$$Q_1(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) = \sum_{m=1}^{M} \sum_{v \in C_m} \left(\sum_{u \in \mathcal{B}_{m,v}} (\bar{\xi}_{m,u,v} \log p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}) + (1 - \bar{\xi}_{m,u,v}) \log(1 - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})) + \sum_{w \in F(v) \setminus C_m} \log(1 - p(\boldsymbol{x}_{v,w}, \boldsymbol{\theta})) \right), \tag{9}$$

$$Q_2(\boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) = \sum_{m=1}^{M} \sum_{v \in C_m} \sum_{u \in \mathcal{B}_{m,v}} (\bar{\mu}_{m,u,v} \log r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi}) - \bar{\xi}_{m,u,v} r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u})), \tag{10}$$

and $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$ is defined by

$$\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) = \sum_{m=1}^{M} \sum_{v \in C_m} \sum_{u \in \mathcal{B}_{m,v}} (\bar{\mu}_{m,u,v} \log \mu_{m,u,v} + (1 - \bar{\mu}_{m,u,v})(\bar{\eta}_{m,u,v} \log \eta_{m,u,v} + (1 - \bar{\eta}_{m,u,v}) \log(1 - \eta_{m,u,v})) \tag{11}$$

Since $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$ is maximized at $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$ and $\boldsymbol{\phi} = \bar{\boldsymbol{\phi}}$ from Equation (11), we can increase the value of $\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi})$ by maximizing $Q(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$ (see Equation (8)).

We can maximize Q by independently maximizing Q_1 and Q_2 with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively. Here, by noting the definition of $p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})$ described in Equation (1), we can derive the gradient vector and the Hessian matrix of Q_1 as follows:

$$\frac{\partial Q_1(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})}{\partial \boldsymbol{\theta}} = \sum_{m=1}^{M} \sum_{v \in C_m} \left(\sum_{u \in \mathcal{B}_{m,v}} (\bar{\xi}_{m,u,v} - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})) \boldsymbol{x}_{u,v} - \sum_{w \in F(v) \setminus C_m} p(\boldsymbol{x}_{v,w}, \boldsymbol{\theta}) \boldsymbol{x}_{v,w} \right), \tag{12}$$

$$\frac{\partial^2 Q_1(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} = -\sum_{m=1}^{M} \sum_{v \in C_m} \left(\sum_{u \in \mathcal{B}_{m,v}} \zeta_{u,v} \boldsymbol{x}_{u,v} \boldsymbol{x}_{u,v}^T + \sum_{w \in F(v) \setminus C_m} \zeta_{v,w} \boldsymbol{x}_{v,w} \boldsymbol{x}_{v,w}^T \right), \tag{13}$$

where $\zeta_{u,v} = p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})(1 - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}))$. We see that the Hessian matrix of Q_1 is non-positive definite, and thus, we can obtain the optimal solution of Q_1 by using the Newton method. Similarly, we can derive the gradient vector and the Hessian matrix of Q_2 as follows:

$$\frac{\partial Q_2(\boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})}{\partial \boldsymbol{\phi}} = \sum_{m=1}^{M} \sum_{v \in C_m} \sum_{u \in \mathcal{B}_{m,v}} (\bar{\mu}_{m,u,v} - \bar{\xi}_{m,u,v} r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u})) \boldsymbol{x}_{u,v}, \tag{14}$$

$$\frac{\partial^2 Q_2(\boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})}{\partial \boldsymbol{\phi} \partial \boldsymbol{\phi}^T} = -\sum_{m=1}^{M} \sum_{v \in C_m} \sum_{u \in \mathcal{B}_{m,v}} \bar{\xi}_{m,u,v} r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u}) \boldsymbol{x}_{u,v} \boldsymbol{x}_{u,v}^T. \tag{15}$$

The Hessian matrix of Q_2 is also non-positive definite, and we can obtain the optimal solution by Q_2. Note that we can regard our estimation method as a variant of the EM

Table 1: Absolute errors of estimated parameter values for each network. Values in parentheses are the assumed true values.

network	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}
Blog	0.0380	0.0587	0.1121	0.0941	0.0874	0.0873	0.0419	0.0723	0.0398	0.0400	0.0378
	(-2.0)	(2.0)	(-1.0)	(0.0)	(0.0)	(0.0)	(1.0)	(-2.0)	(0.0)	(0.0)	(0.0)
Enron	0.0371	0.0465	0.1152	0.0637	0.0758	0.0692	0.0382	0.0831	0.0400	0.0370	0.0385
	(-3.0)	(2.0)	(-1.0)	(0.0)	(0.0)	(0.0)	(1.0)	(-2.0)	(0.0)	(0.0)	(0.0)
Wikipedia	0.0485	0.0455	0.1505	0.0945	0.0710	0.0897	0.0444	0.1079	0.0438	0.0458	0.0434
	(-4.0)	(2.0)	(-1.0)	(0.0)	(0.0)	(0.0)	(1.0)	(-2.0)	(0.0)	(0.0)	(0.0)

algorithm. We want to emphasize here that each time iteration proceeds the value of the likelihood function never decreases and the iterative algorithm is guaranteed to converge due to the convexity of Q.

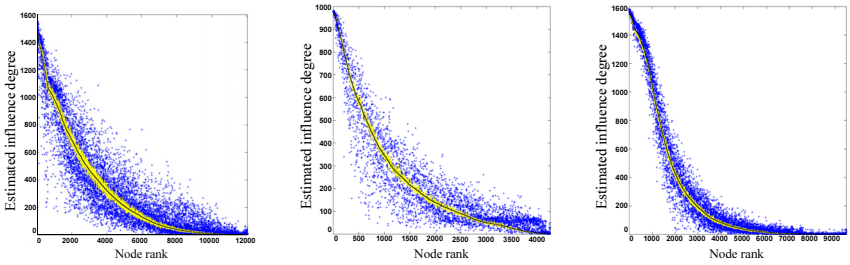
4 Experimental Evaluation

We experimentally evaluated our learning algorithm by using synthetic information diffusion results generated from three large real world networks. Due to the page limitation, here we show only the results for the parameter vector $\boldsymbol{\theta}$, but we observed the similar results for the parameter vector $\boldsymbol{\phi}$. Note that $\boldsymbol{\phi}$ does not affect the influence degree used in our evaluation described later.

4.1 Dataset

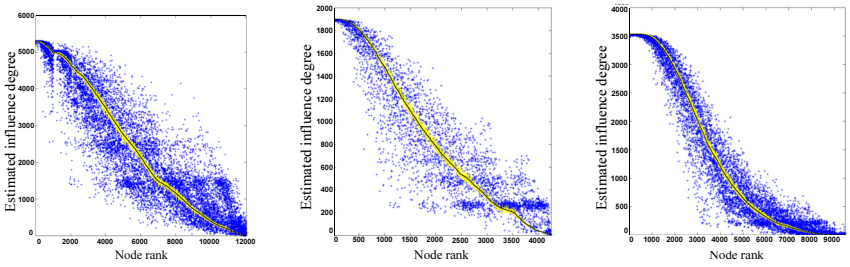
We adopted three datasets of large real networks, which are all bidirectionally connected networks. The first one is a trackback network of Japanese blogs used in [7], and has 12,047 nodes and 79,920 directed links (the blog network). The second one is a network derived from the Enron Email Dataset [11] by extracting the senders and the recipients and linking those that had bidirectional communications and there were 4,254 nodes and 44,314 directed links (the Enron network). The last one is a network of people that was derived from the "list of people" within Japanese Wikipedia, used in [6], which has 9,481 nodes and 245,044 directed links (the Wikipedia network).

For each network, we generated synthetic information diffusion results in the following way: 1) artificially generate node attributes and determine their values in a random manner; 2) determine a parameter vector $\boldsymbol{\theta}$ which is assumed to be true; and then 3) generate 5 distinct information diffusion results, $\mathcal{D}_5 = \{D_1, \cdots, D_5\}$, each of which starts from a randomly selected initial active node, and contains at least 10 active nodes by the AsIC model mentioned in section 2.2. We generated a total of 10 attributes for every node in each network: 5 ordered attributes, each with a non-negative integer less than 20, and 5 nominal attributes, each with either 0, 1, or 2. The true parameter vector $\boldsymbol{\theta}$ was determined so that, according to [5], the average diffusion probability derived from the generated attribute values and $\boldsymbol{\theta}$ becomes smaller than $1/\bar{d}$, where $\bar{d}$ is the mean out-degree of a network. We refer thus determined values to base values. The resulting average diffusion probability was 0.142 for the blog network, 0.062 for the Enron network, and 0.026 for the Wikipedia network, respectively.



(a) Blog (b) Enron (c) Wikipedia

Fig. 1: Comparison of three influence degrees σ (black sold line), $\hat{\sigma}$ (yellow marker) and $\bar{\sigma}$ (blue marker) for one particular run, randomly selected from the 100 independent trials in case that the diffusion probabilities are the base values.



(a) Blog (b) Enron (c) Wikipedia

Fig. 2: Comparison of three influence degrees σ (black sold line), $\hat{\sigma}$ (yellow marker) and $\bar{\sigma}$ (blue marker) for one particular run, randomly selected from the 100 independent trials in case that the diffusion probabilities are larger than the base values.

4.2 Results

First, we examined the accuracy of parameter values $\hat{\boldsymbol{\theta}}$ estimated by our learning algorithm. Table 1 shows the absolute error $|\theta_i - \hat{\theta}_i|$ for each network which is the average over 100 trials, each obtained from a different $\mathcal{D}_5$ [3], where the values in the parentheses are true parameter values. On average, the absolute error of each parameter is 0.0645, 0.0586, and 0.0714 for the blog, Enron, and Wikipedia network, and their standard deviations are 0.0260, 0.0243, and 0.0338, respectively. This result shows that our learning method can estimate parameter values with very high accuracy regardless of networks. Note that $\theta_3, \theta_4, \theta_5, \theta_8, \theta_9$, and θ_{10} are set to 0. This is different from limiting the number of attributes to 4. The average computation time that our learning algorithm spent to estimate the parameter values was 2.96, 6.01, and 28.24 seconds for the blog, Enron, and Wikipedia network, respectively, which means that our learning method is very efficient (machine used is Intel(R) Xeon(R) CPU W5590 @3.33GHz with 32GB memory). Note that, from the derivation in Section 3.2, the computation time depends on the density of the network, i.e. the number of parents of a node.

Next, we evaluated our learning algorithm in terms of the influence degree of each node v which is defined as the expected number of active nodes after the information diffusion is over when v is chosen to be the initial active node. In this experiment,

[3] We generated $\mathcal{D}_5$ 100 times.

we derived the influence degree of each node by computing the empirical mean of the number of active nodes obtained from 1,000 independent runs which are based on the bond percolation technique described in [9]. Here, we compared the influence degree $\hat{\sigma}(v)$ [4] of a node v which was derived using the parameter values estimated by our learning algorithm with the influence degree $\bar{\sigma}(v)$ which was derived by a naive way that uses the uniform diffusion probability approximated by averaging the true link-dependent diffusion probabilities.

Figure 1 presents three influence degrees σ, $\hat{\sigma}$, and $\bar{\sigma}$ for each node v for one particular run, randomly chosen from the 100 independent trials, where σ denotes the influence degree derived using the true link-dependent diffusion probability. The nodes are ordered according to the estimated true rank of influential degree. From these figures, we can observe that the difference between σ (solid line) and $\hat{\sigma}$ (yellow) is quite small, while the difference between σ and $\bar{\sigma}$ (blue) is very large and widely fluctuating. In fact, for $\hat{\sigma}$, the average of the absolute error defined as $|\hat{\sigma}(v) - \sigma(v)|$ over all nodes and all trials is 13.91, 6.80, and 8.32 for the blog, Enron, and Wikipedia network, and their standard deviations are 19.62, 7.98, and 12.96, respectively. Whereas, for $\bar{\sigma}$, the corresponding average of $|\bar{\sigma}(v) - \sigma(v)|$ is 77.75, 54.51, and 35.02, and their standard deviations are 96.13, 57.80, and 51.85, respectively. Even in the best case for $\bar{\sigma}$ (the Wikipedia network), the average error for $\bar{\sigma}$ is about 4 times larger than that for $\hat{\sigma}$.

We further investigated how the error changes with the diffusion probabilities. Figure 2 is the results where the diffusion probabilities are increased, *i.e.*, larger influence degrees expected. To realize this, θ_0 is increased by 1 for each network, i.e. $\theta_0 = -1$, -2, and -3 for the blog, Enron, and Wikipedia network, respectively, which resulted in the corresponding average diffusion probability of 0.28, 0.14, and 0.063, respectively. It is clear that the difference between σ and $\hat{\sigma}$ remains very small, but the difference between σ and $\bar{\sigma}$ becomes larger than before (Fig. 1). Actually, for $\hat{\sigma}$, the average (standard deviation) of the absolute error over all nodes and all trials is 47.95 (37.34), 13.27 (13.36), and 15.11 (17.29) for the blog, Enron, and Wikipedia network, respectively, while, for $\bar{\sigma}$, the corresponding average (standard deviation) is 518.94 (502.08), 162.56 (159.40), and 163.51 (205.18), respectively. These results confirm that $\hat{\sigma}$ remains close to the true influence degree regardless of the diffusion probability p, while $\bar{\sigma}$ is very sensitive to p.

Overall, we can say that our learning algorithm is useful for estimating the influence degrees of nodes in a network, provided that we have some knowledge of dependency of diffusion probability on the selected attributes. It can accurately estimate them from a small amount of information diffusion results and avoid the overfitting problem.

5 Conclusion

Information diffusion over a social network is analyzed by modeling the cascade of interactions of neighboring nodes as probabilistic processes of state changes. The number of the parameters in the model is in general as many as the number of nodes and links, and amounts to several tens of thousands for a network of node size about ten

[4] Note that σ here is not meant to be the standard deviation.

thousands. In this paper, we addressed the problem of estimating link-dependent parameters of probabilistic information diffusion model from a small amount of observed diffusion data. The key idea is not to estimate them directly from the data as has been done in the past studies, but to learn the functional dependency of the parameters on the small number of node attributes. The task is formulated as the maximum likelihood estimation problem, and an efficient parameter update algorithm that guarantees the convergence is derived. We tested the performance of the learning algorithm on three real world networks assuming a particular class of attribute dependency, and confirmed that the dependency can be correctly learned even if the number of parameters (information diffusion probability of each link in this paper) is several tens of thousands. We further showed that the influence degree of each node based on the link-dependent diffusion probabilities is substantially different from that obtained assuming a uniform diffusion probability which is approximated by the average of the link-dependent diffusion probabilities. This indicates that use of uniform diffusion probability is not justified if the true distribution is non-uniform, and affects the influential nodes and their ranking considerably.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Domingos, P.: Mining social networks for viral marketing. IEEE Intell. Syst. 20, 80–82 (2005)
2. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters 12, 211–223 (2001)
3. Gomez-Rodriguez, M., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. In: KDD 2010. pp. 1019–1028 (2010)
4. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. SIGKDD Explorations 6, 43–52 (2004)
5. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: KDD-2003. pp. 137–146 (2003)
6. Kimura, M., Saito, K., Motoda, H.: Minimizing the spread of contamination by blocking links in a network. In: AAAI-08. pp. 1175–1180 (2008)
7. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Trans. Knowl. Discov. Data 3, 9:1–9:23 (2009)
8. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: AAAI-07. pp. 1371–1376 (2007)
9. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Extracting influential nodes on a social network for information diffusion. Data Min. and Knowl. Disc. 20, 70–97 (2010)
10. Kimura, M., Saito, K., Ohara, K., Motoda, H.: Learning to predict opinion share in social networks. In: AAAI-10. pp. 1364–1370 (2010)
11. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: ECML'04. pp. 217–226 (2004)

12. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: EC'06. pp. 228–237 (2006)
13. Myers, S.A., Leskovec, J.: On the convexity of latent social network inference. In: Proceedings of Neural Information Processing Systems (NIPS)
14. Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. 45, 167–256 (2003)
15. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Phys. Rev. E 66, 035101 (2002)
16. Saito, K., Kimura, M., Nakano, R., Motoda, H.: Finding influential nodes in a social network from information diffusion data. In: SBP09. pp. 138–145 (2009)
17. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Learning continuous-time information diffusion model for social behavioral data analysis. In: ACML2009. pp. 322–337 (2009)
18. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Behavioral analyses of information diffusion models by observed data of social network. In: SBP10. pp. 149–158. LNCS 6007 (2010)
19. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Selecting information diffusion models over social networks for behavioral analysis. In: ECML PKDD 2010. pp. 180–195 (2010)
20. Tong, H., Prakash, B.A., Tsoourakakis, C., Eliassi-Rad, T., Faloutsos, C., Chau, D.H.: On the vulnerability of large graphs. In: ICDM 2010. pp. 1091–1096 (2010)
21. Watts, D.J.: A simple model of global cascades on random networks. PNAS 99, 5766–5771 (2002)
22. Watts, D.J., Dodds, P.S.: Influence, networks, and public opinion formation. J. Cons. Res. 34, 441–458 (2007)

Detecting Changes in Opinion Value Distribution for Voter Model

Kazumi Saito[1], Masahiro Kimura[2], Kouzou Ohara[3], and Hiroshi Motoda[4]

[1] School of Administration and Informatics, University of Shizuoka
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
kimura@rins.ryukoku.ac.jp
[3] Department of Integrated Information Technology, Aoyama Gakuin University
ohara@it.aoyama.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We address the problem of detecting the change in opinion share over a social network caused by an unknown external situation change under the value-weighted voter model with multiple opinions in a retrospective setting. The unknown change is treated as a change in the value of an opinion which is a model parameter, and the problem is reduced to detecting this change and its magnitude from the observed opinion share diffusion data. We solved this problem by iteratively maximizing the likelihood of generating the observed opinion share, and in doing so we devised a very efficient search algorithm which avoids parameter value optimization during the search. We tested the performance using the structures of four real world networks and confirmed that the algorithm can efficiently identify the change and outperforms the naive method, in which an exhaustive search is deployed, both in terms of accuracy and computation time.

1 Introduction

Recent technological innovation in the web such as blogosphere and knowledge/media-sharing sites is remarkable, which has made it possible to form various kinds of large social networks, through which behaviors, ideas and opinions can spread, and our behavioral patterns are strongly affected by the interaction with these networks. Thus, substantial attention has been directed to investigating the spread of influence in these networks [9, 2, 14].

Much of the work has treated information as one entity and nodes in the network are either active (influenced) or inactive (uninfluenced), i.e. there are only two states. However, application such as an on-line competitive service in which a user can choose one from multiple choices/decisions requires a model that handles multiple states. In addition, it is important to consider the value of each choice, e.g., quality, brand, authority, etc. because this impacts our choice. We formulated this problem using a value-weighted K opinion diffusion model and provided a way to accurately predict the expected share of the opinions at a future target time from a limited amount of observed data [6]. This model is an extension of the basic voter model which is based on the assumption that a person changes its opinion by the opinions of its neighbors. There has

been a variety of work on the voter model. Dynamical properties of the basic model have been extensively studied including how the degree distribution and the network size affect the mean time to reach consensus [10, 12]. Several variants of the voter model are also investigated and non equilibrium phase transition is analyzed [1, 15]. Yet another line of work extends the voter model by combining it with a network evolution model [3, 2].

These studies are different from what we address in this paper. Almost all of the work so far on information diffusion assumed that the model is stationary. However, our behavior is affected not only by the behaviour of our neighbors but also by other external factors. We apply our voter model to detect a change in opinion share which is caused by an unknown external situation change. We model the change in the external factors as a change in the opinion value, and try to detect the change from the observed opinion share diffusion data. If this is possible, this would bring a substantial advantage. We can detect that something unusual happened during a particular period of time by simply analyzing the data. Note that our approach is retrospective, i.e. we are not predicting the future, but we are trying to understand the phenomena that happened in the past, which shares the same spirit of the work by Kleinberg [7] and Swan [13] in which they tried to organize a huge volume of the data stream and extract structures behind it.

Thus, our problem is reduced to detecting where in time and how long this change persisted and how big this change is. To make the analysis simple, we limit the form of the value change to a rect-linear one, that is, the value changes to a new higher level, persists for a certain period of time and is restored back to the original level and stays the same thereafter. We call this period when the value is high as "hot span" and the rest as "normal span". We use the same parameter optimization algorithm as in [6], i.e. the parameter update algorithm based on the Newton method which globally maximizes the likelihood of generating the observed data sequences. The problem here is more difficult because it has another loop to search for the hot span on top of the above loop. The naive learning algorithm has to iteratively update the patten boundaries (outer loop) and the value must also be optimized for each combination of the pattern boundaries (inner loop), which is extraordinary inefficient. We devised a very efficient search algorithm which avoids the inner loop optimization during the search. We tested the performance using the structures of four real world networks (blog, Wikipedia, Enron and coauthorship), and confirmed that the algorithm can efficiently identify the hot span correctly as well as the opinion value. We further compared our algorithm with the naive method that finds the best combination of change boundaries by an exhaustive search through a set of randomly selected boundary candidates, and showed that the proposed algorithm far outperforms the native method both in terms of accuracy and computation time.

2 Opinion Formation Models

The mathematical model we use for the diffusion of opinions is the value-weighted voter model with K (≥ 2) opinions [6]. A social network is represented by an undirected (bidirectional) graph with self-loops, $G = (V, E)$, where V and E ($\subset V \times V$) are the sets of all the nodes and links in the network, respectively. For a node $v \in V$, let $\Gamma(v)$ denote the set of neighbors of v in G, that is, $\Gamma(v) = \{u \in V; (u, v) \in E\}$. Note that $v \in \Gamma(v)$.

In the model, each node of G is endowed with $(K+1)$ states; opinions $1, \cdots, K$, and *neutral* (i.e., no-opinion state). It is assumed that a node never switches its state from any opinion k to neutral. The model has a parameter w_k (> 0) for each opinion k, which is called the *value-parameter* and must be estimated from observed opinion diffusion data. Let $f_t : V \to \{0, 1, 2, \cdots, K\}$ denote the opinion distribution at time t, where $f_t(v)$ stands for the opinion of node v at time t, and opinion 0 denotes the neutral state. We also denote by $n_k(t, v)$ the number of v's neighbors that hold opinion k at time t for $k = 1, 2, \cdots, K$, i.e., $n_k(t, v) = |\{u \in \Gamma(v);\ f_t(u) = k\}|$. Given a target time T, and an initial state in which each opinion is assigned to only one distinct node and all other nodes are in the neutral state, the evolution process of the model unfolds in the following way. In general, each node v considers changing its opinion based on the current opinions of its neighbors at its $(j-1)$th update-time $t_{j-1}(v)$, and actually changes its opinion at the jth update-time $t_j(v)$, where $t_{j-1}(v) < t_j(v) \leq T$, $j = 1, 2, 3, \cdots$, and $t_0(v) = 0$. It is noted that since node v is included in its neighbors by definition, its own opinion is also reflected. The jth update-time $t_j(v)$ is decided at time $t_{j-1}(v)$ according to the exponential distribution of parameter λ (we simply use $\lambda = 1$ for any $v \in V$)[1]. Then, node v changes its opinion at time $t_j(v)$ as follows: If node v has at least one neighbor with some opinion at time $t_{j-1}(v)$, $f_{t_j(v)}(v) = k$ with probability $w_k n_k(t_{j-1}(v), v) \,/\, \sum_{k'=1}^{K} w_{k'} n_{k'}(t_{j-1}(v), v)$ for $k = 1, \cdots, K$, otherwise, $f_{t_j(v)}(v) = 0$ with probability 1. Note here that $f_t(v) = f_{t_{j-1}(v)}(v)$ for $t_{j-1}(v) \leq t < t_j(v)$. If the next update-time $t_j(v)$ passes T, that is, $t_j(v) > T$, then the opinion evolution of v is over. The evolution process terminates when the opinion evolution of every node in G is over.

Given the observed opinion diffusion data $\mathcal{D}(T_s, T_e) = \{(v, t, f_t(v))\}$ in time-interval $[T_s, T_e]$ (a single example), we consider estimating the values of value-parameters w_1, $\cdots$, w_K, where $0 \leq T_s < T_e \leq T$. From the evolution process of the model, we can obtain the following log likelihood function

$$\mathcal{L}(\boldsymbol{w}; \mathcal{D}(T_s, T_e)) = \log \prod_{(v,t,k) \in C(T_s, T_e)} \frac{n_k(t, v) w_k}{\sum_{k'=1}^{K} n_{k'}(t, v) w_{k'}}, \tag{1}$$

where $\boldsymbol{w} = (w_1, \cdots, w_K)$ stands for the K-dimensional vector of value-parameters, and $C(T_s, T_e) = \{(v, t, f_t(v)) \in \mathcal{D}(T_s, T_e);\ |\{u \in \Gamma(v);\ f_t(u) \neq 0\}| \geq 2\}$. Thus, our estimation problem is formulated as a maximization problem of the objective function $\mathcal{L}(\boldsymbol{w}; \mathcal{D}(T_s, T_e))$ with respect to $\boldsymbol{w}$. We find the optimal values of $\boldsymbol{w}$ by employing a standard Newton method (see [6] for more details).

3 Change Detection Problem

We investigate the problem of detecting the change in behavior of opinion diffusion in a social network G based on the value-weighted voter model with K opinions, which is referred to as the *change detection problem*. In this problem, we assume that some change has happened in the way the opinions diffuse, and we observe the opinion diffusion data in which the change is embedded, and consider detecting where in time and how long this change persisted and how big this change is.

[1] Note that this is equivalent to picking a node randomly and updating its opinion in turn $|V|$ times.

Here, we mathematically formulate the change detection problem. For the opinion diffusion data $\mathcal{D}(0,T)$ in time-interval $[0,T]$, let $[T_1,T_2]$ denote the hot (change) span of the diffusion of opinions. This implies that the intervals $[0,T_1)$ and $(T_2,T]$ are the normal spans. Let $\boldsymbol{w}_n$ and $\boldsymbol{w}_h$ denote the value-parameter vectors for the normal span and the hot span, respectively. Note that $\boldsymbol{w}_n/\|\boldsymbol{w}_n\| \neq \boldsymbol{w}_h/\|\boldsymbol{w}_h\|$ since the opinion dynamics under the value-weighted voter model is invariant to positive scaling of the value-parameter vector $\boldsymbol{w}$, where $\|\boldsymbol{w}_n\|$ and $\|\boldsymbol{w}_h\|$ stand for the norm of vectors $\boldsymbol{w}_n$ and $\boldsymbol{w}_h$. Then, the change detection problem is formulated as follows: Given the opinion diffusion data $\mathcal{D}(0,T)$ in time-interval $[0,T]$, detect the anomalous span $[T_1,T_2]$, and estimate the value-parameter vector $\boldsymbol{w}_h$ of the hot span and the value-parameter vector $\boldsymbol{w}_n$ of the normal span.

Since the value-weighted voter model is a stochastic process model, every sample of opinion diffusion can behave differently. This means that it is quite difficult to accurately detect the true hot span from only a single sample of opinion diffusion. Methods that use only the observed bursty activities, including those proposed by Swan and Allan [13] and Kleinberg [7] would not work. We believe that an explicit use of underlying opinion diffusion model is essential to solve this problem. It is crucially important to detect the hot span precisely in order to identify the external factors which caused the behavioral changes.

4 Detection Methods

4.1 Naive Method

Let $\mathcal{T} = \{t_1, \cdots, t_N\}$ be a set of opinion change time points of all the nodes appearing in the diffusion results $\mathcal{D}(0,T)$. We can consider the following value-parameter vector switching when there is a hot span $S = [T_1,T_2]$:

$$\boldsymbol{w} = \begin{cases} \boldsymbol{w}_n & \text{if } t \in \mathcal{T} \setminus S, \\ \boldsymbol{w}_h & \text{if } t \in \mathcal{T} \cap S. \end{cases}$$

Then, an extended objective function $\mathcal{L}(\boldsymbol{w}_n, \boldsymbol{w}_h; \mathcal{D}(0,T), S)$ can be defined by adequately modifying Equation (1) under this switching scheme. Clearly, the extended objective function is expected to be maximized by setting S to be the true span $S^* = [T_1^*, T_2^*]$, for which $\mathcal{D}(0,T)$ is generated by the value-weighted voter model, provided that $\mathcal{D}(0,T)$ is sufficiently large. Therefore, our hot span detection problem is formalized as the following maximization problem.

$$\hat{S} = \arg\max_{S} \mathcal{L}(\hat{\boldsymbol{w}}_n, \hat{\boldsymbol{w}}_h; \mathcal{D}(0,T), S), \tag{2}$$

where $\hat{\boldsymbol{w}}_n$ and $\hat{\boldsymbol{w}}_h$ denote the maximum likelihood estimators for a given S.

In order to obtain $\hat{S}$ according to Equation (2), we need to prepare a reasonable set of candidate spans, denoted by $\mathcal{S}$. One way of doing so is to construct $\mathcal{S}$ by considering all pairs of observed activation time points. Then, we can construct a set of candidate spans by $\mathcal{S} = \{S = [t_1,t_2] : t_1 < t_2, t_1 \in \mathcal{T}, t_2 \in \mathcal{T}\}$. Equation (2) can be solved by a naive method which has two iterative loops. In the inner loop we first obtain the

maximum likelihood estimators, $\hat{\boldsymbol{w}}_n$ and $\hat{\boldsymbol{w}}_h$, for each candidate S by maximizing the objective function $\mathcal{L}(\boldsymbol{w}_n, \boldsymbol{w}_h; \mathcal{D}(0,T), S)$ using the Newton method. In the outer loop we select the optimal $\hat{S}$ which gives the largest $\mathcal{L}(\hat{\boldsymbol{w}}_n, \hat{\boldsymbol{w}}_h; \mathcal{D}(0,T), S)$ value. However, this method can be extremely inefficient when the number of candidate spans is large. Thus, in order to make it work with a reasonable computational cost, we consider restricting the number of candidate time points to a small value, denoted by J, i.e., we construct $\mathcal{T}_J = \{t_1, \cdots, t_J\}$ by selecting J points from $\mathcal{T}$; then we construct a restricted set of candidate spans by $\mathcal{S}_J = \{S = [t_1, t_2] : t_1 < t_2, t_1 \in \mathcal{T}_J, t_2 \in \mathcal{T}_J\}$. Note that $|\mathcal{S}_J| = J(J-1)/2$, which is large when J is large.

4.2 Proposed Method

It is easily conceivable that the naive method can detect the hot span with a reasonably good accuracy when we set J large at the expense of the computational cost, but the accuracy becomes poorer when we set J smaller to reduce the computational load. We propose a novel detection method below which alleviates this problem and can efficiently and stably detect a hot span from diffusion results $\mathcal{D}(0,T)$.

We first obtain the maximum likelihood estimators, $\hat{\boldsymbol{w}}$ based on the original objective function of Equation (1), and focus on the first-order derivative of the objective function $\mathcal{L}(\boldsymbol{w}; \mathcal{D}(0,T))$ with respect to the value-parameter vector $\boldsymbol{w}$ at each individual opinion change time. More specifically, let $\boldsymbol{w}_t$ be the value-parameter vector at time $t \in \mathcal{T}$. Then we obtain the following formula for the maximum likelihood estimators due to the uniform parameter setting and the globally optimal condition.

$$\frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(0,T))}{\partial \boldsymbol{w}} = \sum_{t \in \mathcal{T}} \frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(0,T))}{\partial \boldsymbol{w}_t} = 0. \tag{3}$$

Now, we can consider the following partial sum for a given hot span $S = [T_1, T_2]$.

$$\boldsymbol{g}(S) = \sum_{t \in \mathcal{T} \cap S} \frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(0,T))}{\partial \boldsymbol{w}_t}. \tag{4}$$

Clearly, $\|\boldsymbol{g}(S)\|$ is likely to have a sufficiently large positive value if $S \approx S^*$ due to our problem setting. Namely, the hot span is detected as follows:

$$\hat{S} = \arg\max_{S \in \mathcal{S}} \|\boldsymbol{g}(S)\|. \tag{5}$$

Here note that we can incrementally calculate $\boldsymbol{g}(S)$. More specifically, let $\mathcal{T} = \{t_1, \cdots, t_N\}$ be a set of candidate time points, where $t_i < t_j$ if $i < j$; then, we can obtain the following formula.

$$\boldsymbol{g}([t_i, t_{j+1}]) = \boldsymbol{g}([t_i, t_j]) + \frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(0,T))}{\partial \boldsymbol{w}_{t_{j+1}}}. \tag{6}$$

The computational cost of the proposed method for examining each candidate span is much smaller than the naive method described above. When $|\mathcal{T}| = N$ is very large, we construct a restricted set of candidate spans $\mathcal{S}_J$ as explained above. We summarize our proposed method below.

1. Maximize $\mathcal{L}(\boldsymbol{w}; \mathcal{D}(0,T))$ by using the Newton method.
2. Construct the candidate time set $\mathcal{T}$ and the candidate span set $\mathcal{S}$.
3. Detect a hot span $\hat{S}$ by Equation (5) and output $\hat{S}$.
4. Maximize $\mathcal{L}(\boldsymbol{w}_n, \boldsymbol{w}_h; \mathcal{D}(0,T), \hat{S})$ by using the Newton method, and output $(\hat{\boldsymbol{w}}_n, \hat{\boldsymbol{w}}_h)$.

Here note that the proposed method requires likelihood maximization by using the Newton method only twice.

5 Experimental Evaluation

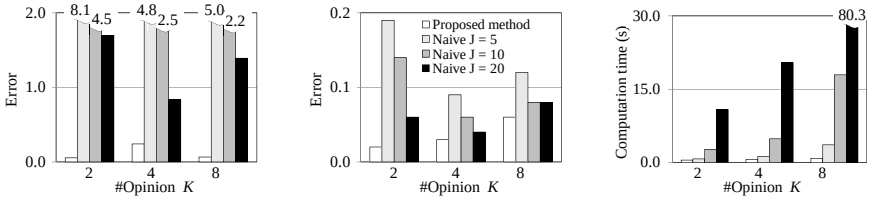
We adopted four datasets of large real networks. They are all bidirectionally connected networks. The first one is a trackback network of Japanese blogs used in [5], which has $12,047$ nodes and $79,920$ directed links (the blog network). The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia, used in [4], and has $9,481$ nodes and $245,044$ directed links (the Wikipedia network). The third one is a network derived from the Enron Email Dataset [8] by extracting the senders and the recipients and linking those that had bidirectional communications. It has $4,254$ nodes and $44,314$ directed links (the Enron network). The fourth one is a coauthorship network used in [11], which has $12,357$ nodes and $38,896$ directed links (the coauthorship network).

For each of these networks, we generated opinion diffusion results for three different values of K (the number of opinions), i.e., $K = 2$, 4, and 8, by choosing the top K nodes with respect to node degree ranking as the initial K nodes and simulating the model mentioned in section 2 from 0 to $T = 25$. We assumed that the value of all the opinions were initially 1.0, i.e. the value-parameters for all the opinions are 1.0 for the normal span, and further assumed that the value of the first opinion changed to double for a period of $[10, 15]$, i.e. the value-parameter of the fast opinion is 2.0 and the value-parameters of all the other opinions are 1.0 for the hot span. We then estimated the hot span and the value-parameters for both the spans (normal and hot) by the two methods (the proposed and the naive), and compared their accuracy and the computation time. We adopted $1,000$ as the value of J (the number of candidate time points) for the proposed method, and 5, 10, and 20 for the naive method.

Figures 1 and 2 show the experimental results[2] where each value is the average over 10 trials for 10 distinct diffusion results. We evaluated the accuracy of the estimated hot span $[\hat{T}_1, \hat{T}_2]$ by the absolute error $|\hat{T}_1 - T_1| + |\hat{T}_2 - T_2|$, and the accuracy of the estimated opinion values $\hat{\boldsymbol{w}}$ by the mean absolute error $\Sigma_{i=1}^{K}(|\hat{w}_{in} - w_{in}| + |\hat{w}_{ih} - w_{ih}|)/K$, where w_{in} and w_{ih} are values of opinion i for the normal and the hot spans, respectively.

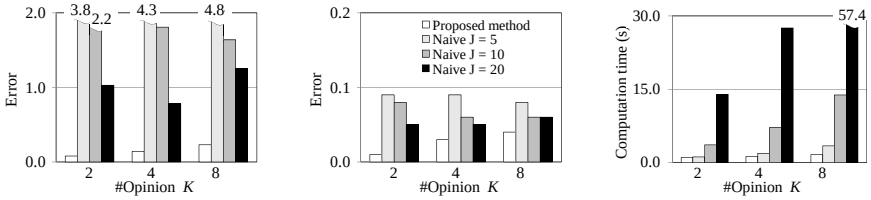
From these results, we can find that the proposed method is much more accurate than the naive method for both the networks. The average error for the naive method decreases as J becomes larger. But, even in the best case for the naive method ($J = 20$), its average error in the estimation of the hot span is maximum about 30 times larger than that of proposed method (in the case of the Enron network under $K = 2$), and it is maximum about 6 times larger in the estimation of value-parameters (in the case of

[2] We only show the results for the two networks (Enron and coauthorship) due to the space limitation. In fact, we obtained similar results also for the other two networks (blog and Wikipedia).



(a) Error of $[\hat{T}_1, \hat{T}_2]$ (b) Error of $\hat{\boldsymbol{w}}$ (c) Computation time

Fig. 1: Comparison on the Enron network



(a) Error of $[\hat{T}_1, \hat{T}_2]$ (b) Error of $\hat{\boldsymbol{w}}$ (c) Computation time

Fig. 2: Comparison on the coauthorship network

the coauthorship network under $K = 2$). It is noted that the naive method needs much longer computation time to achieve these best accuracies than the proposed method although the number of candidate time points for the naive method is 50 times smaller. Indeed, it is about 20 times longer for the former case, about 13 times longer for the latter case, and maximum about 95 times longer for the whole results (in the case of the Enron network under $K = 8$). From these results, it can be concluded that the proposed method is able to detect and estimate the hot span and value-parameters much more accurately and efficiently compared with the naive method.

6 Conclusions

In this paper, we addressed the problem of detecting the unusual change in opinion share from the observed data in a retrospective setting, assuming that the opinion share evolves by the value-weighted voter model with multiple opinions. We defined the hot span as the period during which the value of an opinion is changed to a higher value than the other periods which are defined as the normal spans. A naive method to detect such a hot span would iteratively update the pattern boundaries that form a hot span (outer loop) and iteratively update the opinion value for each hot span candidate (inner loop) such that the likelihood function is maximized. This is very inefficient and totally unacceptable. We developed a novel method that avoids the inner loop optimization during search. It only needs to estimate the value twice by the iterative updating algorithm (Newton method), which can reduce the computation times by 7 to 95 times, and is very efficient. We applied the proposed method to opinion share samples generated from four real world large networks and compared the performance with the naive method that considers only the randomly selected boundary candidates. The results clearly indicate that the proposed method far outperforms the naive method both

in terms of accuracy and efficiency. Although we assumed a simplified problem setting in this paper, the proposed method can be easily extended to solve more intricate problems. As the future work, we plan to extend this framework to spatio-temporal hot span detection problems.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Castellano, C., Munoz, M.A., Pastor-Satorras, R.: Nonlinear *q*-voter model. Physical Review E 80, 041129 (2009)
2. Crandall, D., Cosley, D., Huttenlocner, D., Kleinberg, J., Suri, S.: Feedback effects between similarity and sociai infiuence in online communities. In: Proceedings of KDD 2008. pp. 160–168 (2008)
3. Holme, P., Newman, M.E.J.: Nonequilibrium phase transition in the coevolution of networks and opinions. Physical Review E 74, 056108 (2006)
4. Kimura, M., Saito, K., Motoda, H.: Minimizing the spread of contamination by blocking links in a network. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08). pp. 1175–1180 (2008)
5. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data 3, 9:1–9:23 (2009)
6. Kimura, M., Saito, K., Ohara, K., Motoda, H.: Learning to predict opinion share in social networks. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-10). pp. 1364–1370 (2010)
7. Kleinberg, J.: Bursty and hierarchical structure in streams. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002). pp. 91–101 (2002)
8. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: Proceedings of the 2004 European Conference on Machine Learning (ECML'04). pp. 217–226 (2004)
9. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06). pp. 228–237 (2006)
10. Liggett, T.M.: Stochastic interacting systems: contact, voter, and exclusion processes. Spriger, New York (1999)
11. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
12. Sood, V., Redner, S.: Voter model on heterogeneous graphs. Physical Review Letters 94, 178701 (2005)
13. Swan, R., Allan, J.: Automatic generation of overview timelines. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000). pp. 49–56 (2000)
14. Wu, F., Huberman, B.A.: How public opinion forms. In: Proceedings of WINE 2008. pp. 334–341 (2008)
15. Yang, H., Wu, Z., Zhou, C., Zhou, T., Wang, B.: Effects of social diversity on the emergence of global consensus in opinion dynamics. Physical Review E 80, 046108 (2009)

Detecting Changes in Opinion Value Distribution for Voter Model

Kazumi Saito[1], Masahiro Kimura[2], Kouzou Ohara[3], and Hiroshi Motoda[4]

[1] School of Administration and Informatics, University of Shizuoka
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
kimura@rins.ryukoku.ac.jp
[3] Department of Integrated Information Technology, Aoyama Gakuin University
ohara@it.aoyama.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We address the problem of detecting the change in opinion share over a social network caused by an unknown external situation change under the value-weighted voter model with multiple opinions in a retrospective setting. The unknown change is treated as a change in the value of an opinion which is a model parameter, and the problem is reduced to detecting this change and its magnitude from the observed opinion share diffusion data. We solved this problem by iteratively maximizing the likelihood of generating the observed opinion share, and in doing so we devised a very efficient search algorithm which avoids parameter value optimization during the search. We tested the performance using the structures of four real world networks and confirmed that the algorithm can efficiently identify the change and outperforms the naive method, in which an exhaustive search is deployed, both in terms of accuracy and computation time.

1 Introduction

Recent technological innovation in the web such as blogosphere and knowledge/media-sharing sites is remarkable, which has made it possible to form various kinds of large social networks, through which behaviors, ideas and opinions can spread, and our behavioral patterns are strongly affected by the interaction with these networks. Thus, substantial attention has been directed to investigating the spread of influence in these networks [9, 2, 14].

Much of the work has treated information as one entity and nodes in the network are either active (influenced) or inactive (uninfluenced), i.e. there are only two states. However, application such as an on-line competitive service in which a user can choose one from multiple choices/decisions requires a model that handles multiple states. In addition, it is important to consider the value of each choice, e.g., quality, brand, authority, etc. because this impacts our choice. We formulated this problem using a value-weighted K opinion diffusion model and provided a way to accurately predict the expected share of the opinions at a future target time from a limited amount of observed data [6]. This model is an extension of the basic voter model which is based on the assumption that a person changes its opinion by the opinions of its neighbors. There has

been a variety of work on the voter model. Dynamical properties of the basic model have been extensively studied including how the degree distribution and the network size affect the mean time to reach consensus [10, 12]. Several variants of the voter model are also investigated and non equilibrium phase transition is analyzed [1, 15]. Yet another line of work extends the voter model by combining it with a network evolution model [3, 2].

These studies are different from what we address in this paper. Almost all of the work so far on information diffusion assumed that the model is stationary. However, our behavior is affected not only by the behaviour of our neighbors but also by other external factors. We apply our voter model to detect a change in opinion share which is caused by an unknown external situation change. We model the change in the external factors as a change in the opinion value, and try to detect the change from the observed opinion share diffusion data. If this is possible, this would bring a substantial advantage. We can detect that something unusual happened during a particular period of time by simply analyzing the data. Note that our approach is retrospective, i.e. we are not predicting the future, but we are trying to understand the phenomena that happened in the past, which shares the same spirit of the work by Kleinberg [7] and Swan [13] in which they tried to organize a huge volume of the data stream and extract structures behind it.

Thus, our problem is reduced to detecting where in time and how long this change persisted and how big this change is. To make the analysis simple, we limit the form of the value change to a rect-linear one, that is, the value changes to a new higher level, persists for a certain period of time and is restored back to the original level and stays the same thereafter. We call this period when the value is high as "hot span" and the rest as "normal span". We use the same parameter optimization algorithm as in [6], i.e. the parameter update algorithm based on the Newton method which globally maximizes the likelihood of generating the observed data sequences. The problem here is more difficult because it has another loop to search for the hot span on top of the above loop. The naive learning algorithm has to iteratively update the patten boundaries (outer loop) and the value must also be optimized for each combination of the pattern boundaries (inner loop), which is extraordinary inefficient. We devised a very efficient search algorithm which avoids the inner loop optimization during the search. We tested the performance using the structures of four real world networks (blog, Wikipedia, Enron and coauthorship), and confirmed that the algorithm can efficiently identify the hot span correctly as well as the opinion value. We further compared our algorithm with the naive method that finds the best combination of change boundaries by an exhaustive search through a set of randomly selected boundary candidates, and showed that the proposed algorithm far outperforms the native method both in terms of accuracy and computation time.

2 Opinion Formation Models

The mathematical model we use for the diffusion of opinions is the value-weighted voter model with K (≥ 2) opinions [6]. A social network is represented by an undirected (bidirectional) graph with self-loops, $G = (V, E)$, where V and E ($\subset V \times V$) are the sets of all the nodes and links in the network, respectively. For a node $v \in V$, let $\Gamma(v)$ denote the set of neighbors of v in G, that is, $\Gamma(v) = \{u \in V; (u, v) \in E\}$. Note that $v \in \Gamma(v)$.

In the model, each node of G is endowed with $(K+1)$ states; opinions $1, \cdots, K$, and *neutral* (i.e., no-opinion state). It is assumed that a node never switches its state from any opinion k to neutral. The model has a parameter w_k (> 0) for each opinion k, which is called the *value-parameter* and must be estimated from observed opinion diffusion data. Let $f_t : V \to \{0, 1, 2, \cdots, K\}$ denote the opinion distribution at time t, where $f_t(v)$ stands for the opinion of node v at time t, and opinion 0 denotes the neutral state. We also denote by $n_k(t, v)$ the number of v's neighbors that hold opinion k at time t for $k = 1, 2, \cdots, K$, i.e., $n_k(t, v) = |\{u \in \Gamma(v);\ f_t(u) = k\}|$. Given a target time T, and an initial state in which each opinion is assigned to only one distinct node and all other nodes are in the neutral state, the evolution process of the model unfolds in the following way. In general, each node v considers changing its opinion based on the current opinions of its neighbors at its $(j-1)$th update-time $t_{j-1}(v)$, and actually changes its opinion at the jth update-time $t_j(v)$, where $t_{j-1}(v) < t_j(v) \leq T$, $j = 1, 2, 3, \cdots$, and $t_0(v) = 0$. It is noted that since node v is included in its neighbors by definition, its own opinion is also reflected. The jth update-time $t_j(v)$ is decided at time $t_{j-1}(v)$ according to the exponential distribution of parameter λ (we simply use $\lambda = 1$ for any $v \in V$)[1]. Then, node v changes its opinion at time $t_j(v)$ as follows: If node v has at least one neighbor with some opinion at time $t_{j-1}(v)$, $f_{t_j(v)}(v) = k$ with probability $w_k n_k(t_{j-1}(v), v) \,/\, \sum_{k'=1}^{K} w_{k'} n_{k'}(t_{j-1}(v), v)$ for $k = 1, \cdots, K$, otherwise, $f_{t_j(v)}(v) = 0$ with probability 1. Note here that $f_t(v) = f_{t_{j-1}(v)}(v)$ for $t_{j-1}(v) \leq t < t_j(v)$. If the next update-time $t_j(v)$ passes T, that is, $t_j(v) > T$, then the opinion evolution of v is over. The evolution process terminates when the opinion evolution of every node in G is over.

Given the observed opinion diffusion data $\mathcal{D}(T_s, T_e) = \{(v, t, f_t(v))\}$ in time-interval $[T_s, T_e]$ (a single example), we consider estimating the values of value-parameters w_1, $\cdots$, w_K, where $0 \leq T_s < T_e \leq T$. From the evolution process of the model, we can obtain the following log likelihood function

$$\mathcal{L}(\boldsymbol{w}; \mathcal{D}(T_s, T_e)) = \log \prod_{(v,t,k) \in C(T_s, T_e)} \frac{n_k(t, v) w_k}{\sum_{k'=1}^{K} n_{k'}(t, v) w_{k'}}, \tag{1}$$

where $\boldsymbol{w} = (w_1, \cdots, w_K)$ stands for the K-dimensional vector of value-parameters, and $C(T_s, T_e) = \{(v, t, f_t(v)) \in \mathcal{D}(T_s, T_e);\ |\{u \in \Gamma(v);\ f_t(u) \neq 0\}| \geq 2\}$. Thus, our estimation problem is formulated as a maximization problem of the objective function $\mathcal{L}(\boldsymbol{w}; \mathcal{D}(T_s, T_e))$ with respect to $\boldsymbol{w}$. We find the optimal values of $\boldsymbol{w}$ by employing a standard Newton method (see [6] for more details).

3 Change Detection Problem

We investigate the problem of detecting the change in behavior of opinion diffusion in a social network G based on the value-weighted voter model with K opinions, which is referred to as the *change detection problem*. In this problem, we assume that some change has happened in the way the opinions diffuse, and we observe the opinion diffusion data in which the change is embedded, and consider detecting where in time and how long this change persisted and how big this change is.

[1] Note that this is equivalent to picking a node randomly and updating its opinion in turn $|V|$ times.

Here, we mathematically formulate the change detection problem. For the opinion diffusion data $\mathcal{D}(0,T)$ in time-interval $[0,T]$, let $[T_1,T_2]$ denote the hot (change) span of the diffusion of opinions. This implies that the intervals $[0,T_1)$ and $(T_2,T]$ are the normal spans. Let $\boldsymbol{w}_n$ and $\boldsymbol{w}_h$ denote the value-parameter vectors for the normal span and the hot span, respectively. Note that $\boldsymbol{w}_n/\|\boldsymbol{w}_n\| \neq \boldsymbol{w}_h/\|\boldsymbol{w}_h\|$ since the opinion dynamics under the value-weighted voter model is invariant to positive scaling of the value-parameter vector $\boldsymbol{w}$, where $\|\boldsymbol{w}_n\|$ and $\|\boldsymbol{w}_h\|$ stand for the norm of vectors $\boldsymbol{w}_n$ and $\boldsymbol{w}_h$. Then, the change detection problem is formulated as follows: Given the opinion diffusion data $\mathcal{D}(0,T)$ in time-interval $[0,T]$, detect the anomalous span $[T_1,T_2]$, and estimate the value-parameter vector $\boldsymbol{w}_h$ of the hot span and the value-parameter vector $\boldsymbol{w}_n$ of the normal span.

Since the value-weighted voter model is a stochastic process model, every sample of opinion diffusion can behave differently. This means that it is quite difficult to accurately detect the true hot span from only a single sample of opinion diffusion. Methods that use only the observed bursty activities, including those proposed by Swan and Allan [13] and Kleinberg [7] would not work. We believe that an explicit use of underlying opinion diffusion model is essential to solve this problem. It is crucially important to detect the hot span precisely in order to identify the external factors which caused the behavioral changes.

4 Detection Methods

4.1 Naive Method

Let $\mathcal{T} = \{t_1, \cdots, t_N\}$ be a set of opinion change time points of all the nodes appearing in the diffusion results $\mathcal{D}(0,T)$. We can consider the following value-parameter vector switching when there is a hot span $S = [T_1,T_2]$:

$$\boldsymbol{w} = \begin{cases} \boldsymbol{w}_n & \text{if } t \in \mathcal{T} \setminus S, \\ \boldsymbol{w}_h & \text{if } t \in \mathcal{T} \cap S. \end{cases}$$

Then, an extended objective function $\mathcal{L}(\boldsymbol{w}_n, \boldsymbol{w}_h; \mathcal{D}(0,T), S)$ can be defined by adequately modifying Equation (1) under this switching scheme. Clearly, the extended objective function is expected to be maximized by setting S to be the true span $S^* = [T_1^*, T_2^*]$, for which $\mathcal{D}(0,T)$ is generated by the value-weighted voter model, provided that $\mathcal{D}(0,T)$ is sufficiently large. Therefore, our hot span detection problem is formalized as the following maximization problem.

$$\hat{S} = \arg\max_{S} \mathcal{L}(\hat{\boldsymbol{w}}_n, \hat{\boldsymbol{w}}_h; \mathcal{D}(0,T), S), \tag{2}$$

where $\hat{\boldsymbol{w}}_n$ and $\hat{\boldsymbol{w}}_h$ denote the maximum likelihood estimators for a given S.

In order to obtain $\hat{S}$ according to Equation (2), we need to prepare a reasonable set of candidate spans, denoted by $\mathcal{S}$. One way of doing so is to construct $\mathcal{S}$ by considering all pairs of observed activation time points. Then, we can construct a set of candidate spans by $\mathcal{S} = \{S = [t_1,t_2] : t_1 < t_2, t_1 \in \mathcal{T}, t_2 \in \mathcal{T}\}$. Equation (2) can be solved by a naive method which has two iterative loops. In the inner loop we first obtain the

maximum likelihood estimators, $\hat{\boldsymbol{w}}_n$ and $\hat{\boldsymbol{w}}_h$, for each candidate S by maximizing the objective function $\mathcal{L}(\boldsymbol{w}_n, \boldsymbol{w}_h; \mathcal{D}(0,T), S)$ using the Newton method. In the outer loop we select the optimal $\hat{S}$ which gives the largest $\mathcal{L}(\hat{\boldsymbol{w}}_n, \hat{\boldsymbol{w}}_h; \mathcal{D}(0,T), S)$ value. However, this method can be extremely inefficient when the number of candidate spans is large. Thus, in order to make it work with a reasonable computational cost, we consider restricting the number of candidate time points to a small value, denoted by J, i.e., we construct $\mathcal{T}_J = \{t_1, \cdots, t_J\}$ by selecting J points from $\mathcal{T}$; then we construct a restricted set of candidate spans by $\mathcal{S}_J = \{S = [t_1, t_2] : t_1 < t_2, t_1 \in \mathcal{T}_J, t_2 \in \mathcal{T}_J\}$. Note that $|\mathcal{S}_J| = J(J-1)/2$, which is large when J is large.

4.2 Proposed Method

It is easily conceivable that the naive method can detect the hot span with a reasonably good accuracy when we set J large at the expense of the computational cost, but the accuracy becomes poorer when we set J smaller to reduce the computational load. We propose a novel detection method below which alleviates this problem and can efficiently and stably detect a hot span from diffusion results $\mathcal{D}(0,T)$.

We first obtain the maximum likelihood estimators, $\hat{\boldsymbol{w}}$ based on the original objective function of Equation (1), and focus on the first-order derivative of the objective function $\mathcal{L}(\boldsymbol{w}; \mathcal{D}(0,T))$ with respect to the value-parameter vector $\boldsymbol{w}$ at each individual opinion change time. More specifically, let $\boldsymbol{w}_t$ be the value-parameter vector at time $t \in \mathcal{T}$. Then we obtain the following formula for the maximum likelihood estimators due to the uniform parameter setting and the globally optimal condition.

$$\frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(0,T))}{\partial \boldsymbol{w}} = \sum_{t \in \mathcal{T}} \frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(0,T))}{\partial \boldsymbol{w}_t} = 0. \tag{3}$$

Now, we can consider the following partial sum for a given hot span $S = [T_1, T_2]$.

$$\boldsymbol{g}(S) = \sum_{t \in \mathcal{T} \cap S} \frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(0,T))}{\partial \boldsymbol{w}_t}. \tag{4}$$

Clearly, $\|\boldsymbol{g}(S)\|$ is likely to have a sufficiently large positive value if $S \approx S^*$ due to our problem setting. Namely, the hot span is detected as follows:

$$\hat{S} = \arg\max_{S \in \mathcal{S}} \|\boldsymbol{g}(S)\|. \tag{5}$$

Here note that we can incrementally calculate $\boldsymbol{g}(S)$. More specifically, let $\mathcal{T} = \{t_1, \cdots, t_N\}$ be a set of candidate time points, where $t_i < t_j$ if $i < j$; then, we can obtain the following formula.

$$\boldsymbol{g}([t_i, t_{j+1}]) = \boldsymbol{g}([t_i, t_j]) + \frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(0,T))}{\partial \boldsymbol{w}_{t_{j+1}}}. \tag{6}$$

The computational cost of the proposed method for examining each candidate span is much smaller than the naive method described above. When $|\mathcal{T}| = N$ is very large, we construct a restricted set of candidate spans $\mathcal{S}_J$ as explained above. We summarize our proposed method below.

1. Maximize $\mathcal{L}(\boldsymbol{w}; \mathcal{D}(0,T))$ by using the Newton method.
2. Construct the candidate time set $\mathcal{T}$ and the candidate span set $\mathcal{S}$.
3. Detect a hot span $\hat{S}$ by Equation (5) and output $\hat{S}$.
4. Maximize $\mathcal{L}(\boldsymbol{w}_n, \boldsymbol{w}_h; \mathcal{D}(0,T), \hat{S})$ by using the Newton method, and output $(\hat{\boldsymbol{w}}_n, \hat{\boldsymbol{w}}_h)$.

Here note that the proposed method requires likelihood maximization by using the Newton method only twice.

5 Experimental Evaluation

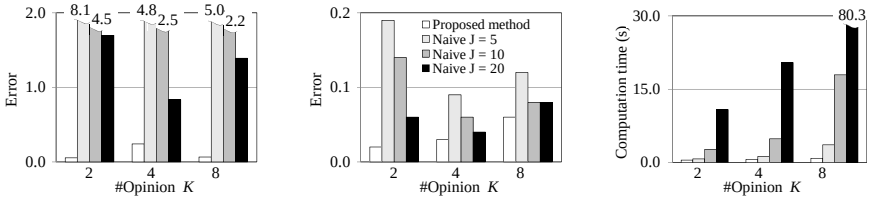
We adopted four datasets of large real networks. They are all bidirectionally connected networks. The first one is a trackback network of Japanese blogs used in [5], which has $12,047$ nodes and $79,920$ directed links (the blog network). The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia, used in [4], and has $9,481$ nodes and $245,044$ directed links (the Wikipedia network). The third one is a network derived from the Enron Email Dataset [8] by extracting the senders and the recipients and linking those that had bidirectional communications. It has $4,254$ nodes and $44,314$ directed links (the Enron network). The fourth one is a coauthorship network used in [11], which has $12,357$ nodes and $38,896$ directed links (the coauthorship network).

For each of these networks, we generated opinion diffusion results for three different values of K (the number of opinions), i.e., $K = 2$, 4, and 8, by choosing the top K nodes with respect to node degree ranking as the initial K nodes and simulating the model mentioned in section 2 from 0 to $T = 25$. We assumed that the value of all the opinions were initially 1.0, i.e. the value-parameters for all the opinions are 1.0 for the normal span, and further assumed that the value of the first opinion changed to double for a period of $[10, 15]$, i.e. the value-parameter of the fast opinion is 2.0 and the value-parameters of all the other opinions are 1.0 for the hot span. We then estimated the hot span and the value-parameters for both the spans (normal and hot) by the two methods (the proposed and the naive), and compared their accuracy and the computation time. We adopted $1,000$ as the value of J (the number of candidate time points) for the proposed method, and 5, 10, and 20 for the naive method.

Figures 1 and 2 show the experimental results[2] where each value is the average over 10 trials for 10 distinct diffusion results. We evaluated the accuracy of the estimated hot span $[\hat{T}_1, \hat{T}_2]$ by the absolute error $|\hat{T}_1 - T_1| + |\hat{T}_2 - T_2|$, and the accuracy of the estimated opinion values $\hat{\boldsymbol{w}}$ by the mean absolute error $\Sigma_{i=1}^{K}(|\hat{w}_{in} - w_{in}| + |\hat{w}_{ih} - w_{ih}|)/K$, where w_{in} and w_{ih} are values of opinion i for the normal and the hot spans, respectively.

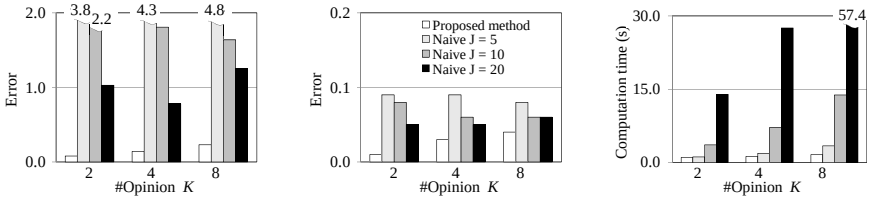
From these results, we can find that the proposed method is much more accurate than the naive method for both the networks. The average error for the naive method decreases as J becomes larger. But, even in the best case for the naive method ($J = 20$), its average error in the estimation of the hot span is maximum about 30 times larger than that of proposed method (in the case of the Enron network under $K = 2$), and it is maximum about 6 times larger in the estimation of value-parameters (in the case of

[2] We only show the results for the two networks (Enron and coauthorship) due to the space limitation. In fact, we obtained similar results also for the other two networks (blog and Wikipedia).



(a) Error of $[\hat{T}_1, \hat{T}_2]$ (b) Error of $\hat{\boldsymbol{w}}$ (c) Computation time

Fig. 1: Comparison on the Enron network



(a) Error of $[\hat{T}_1, \hat{T}_2]$ (b) Error of $\hat{\boldsymbol{w}}$ (c) Computation time

Fig. 2: Comparison on the coauthorship network

the coauthorship network under $K = 2$). It is noted that the naive method needs much longer computation time to achieve these best accuracies than the proposed method although the number of candidate time points for the naive method is 50 times smaller. Indeed, it is about 20 times longer for the former case, about 13 times longer for the latter case, and maximum about 95 times longer for the whole results (in the case of the Enron network under $K = 8$). From these results, it can be concluded that the proposed method is able to detect and estimate the hot span and value-parameters much more accurately and efficiently compared with the naive method.

6 Conclusions

In this paper, we addressed the problem of detecting the unusual change in opinion share from the observed data in a retrospective setting, assuming that the opinion share evolves by the value-weighted voter model with multiple opinions. We defined the hot span as the period during which the value of an opinion is changed to a higher value than the other periods which are defined as the normal spans. A naive method to detect such a hot span would iteratively update the pattern boundaries that form a hot span (outer loop) and iteratively update the opinion value for each hot span candidate (inner loop) such that the likelihood function is maximized. This is very inefficient and totally unacceptable. We developed a novel method that avoids the inner loop optimization during search. It only needs to estimate the value twice by the iterative updating algorithm (Newton method), which can reduce the computation times by 7 to 95 times, and is very efficient. We applied the proposed method to opinion share samples generated from four real world large networks and compared the performance with the naive method that considers only the randomly selected boundary candidates. The results clearly indicate that the proposed method far outperforms the naive method both

in terms of accuracy and efficiency. Although we assumed a simplified problem setting in this paper, the proposed method can be easily extended to solve more intricate problems. As the future work, we plan to extend this framework to spatio-temporal hot span detection problems.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Castellano, C., Munoz, M.A., Pastor-Satorras, R.: Nonlinear *q*-voter model. Physical Review E 80, 041129 (2009)
2. Crandall, D., Cosley, D., Huttenlocner, D., Kleinberg, J., Suri, S.: Feedback effects between similarity and sociai infiuence in online communities. In: Proceedings of KDD 2008. pp. 160–168 (2008)
3. Holme, P., Newman, M.E.J.: Nonequilibrium phase transition in the coevolution of networks and opinions. Physical Review E 74, 056108 (2006)
4. Kimura, M., Saito, K., Motoda, H.: Minimizing the spread of contamination by blocking links in a network. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08). pp. 1175–1180 (2008)
5. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data 3, 9:1–9:23 (2009)
6. Kimura, M., Saito, K., Ohara, K., Motoda, H.: Learning to predict opinion share in social networks. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-10). pp. 1364–1370 (2010)
7. Kleinberg, J.: Bursty and hierarchical structure in streams. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002). pp. 91–101 (2002)
8. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: Proceedings of the 2004 European Conference on Machine Learning (ECML'04). pp. 217–226 (2004)
9. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06). pp. 228–237 (2006)
10. Liggett, T.M.: Stochastic interacting systems: contact, voter, and exclusion processes. Spriger, New York (1999)
11. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
12. Sood, V., Redner, S.: Voter model on heterogeneous graphs. Physical Review Letters 94, 178701 (2005)
13. Swan, R., Allan, J.: Automatic generation of overview timelines. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000). pp. 49–56 (2000)
14. Wu, F., Huberman, B.A.: How public opinion forms. In: Proceedings of WINE 2008. pp. 334–341 (2008)
15. Yang, H., Wu, Z., Zhou, C., Zhou, T., Wang, B.: Effects of social diversity on the emergence of global consensus in opinion dynamics. Physical Review E 80, 046108 (2009)

Learning Information Diffusion Model in a Social Network for Predicting Influence of Nodes

Masahiro Kimura (*corresponding author)

Department of Electronics and Informatics

Ryukoku University

Seta, Otsu 520-2194, Japan

Tel: +81 77 543 7406

Fax: +81 77 543 7428

Email: kimura@rins.ryukoku.ac.jp

Kazumi Saito

School of Administration and Informatics

University of Shizuoka

Shizuoka 422-8526, Japan

Kouzou Ohara

Department of Integrated Information Technology

Aoyama Gakuin University

Kanagawa 229-8558, Japan

Hiroshi Motoda

Institute of Scientific and Industrial Research

Osaka University

Osaka 567-0047, Japan

Abstract

We address the problem of estimating the parameters, from observed data in a complex social network, for an information diffusion model that takes time-delay into account, based on the popular independent cascade (IC) model. For this purpose we formulate the likelihood to obtain the observed data which is a set of time-sequence data of infected (active) nodes, and propose an iterative method to search for the parameters (time-delay and diffusion) that maximize this likelihood. We first show by using a synthetic network that the proposed method outperforms the similar existing method. Next, we apply this method to problems of both 1) predicting the influence of nodes for the considered information diffusion model and 2) ranking the influential nodes. Using three large social networks, we demonstrate the effectiveness of the proposed method.

Keywords

Social network analysis, information diffusion model with time-delay, parameter learning, influence degree prediction, node ranking

1 Introduction

Investigating the structure and function of complex networks such as biochemical and social networks is a hot research subject [2, 15, 5]. From a functional view, innovation, topics and even malicious rumors can propagate through social networks among people in the form of so-called "word-of-mouth" communications. Therefore, considerable attention has recently been devoted to social networks as an important medium for the spread of information [14, 7, 4, 13, 21].

There are several models that simulate information diffusion through a network. A widely-used fundamental probabilistic model is the *independent cascade (IC) model* [6, 8], which can be regarded as the so-called *susceptible/infected/recovered (SIR) model* for the spread of a disease [15]. This model has been used to solve the problem of finding a limited number of nodes that are influential for the spread of information [8, 9]. This combinatorial optimization problem is called the *influence maximization problem*, and is one of the important application problems in sociology and "viral marketing" [1]. Further, this model has been used to solve yet another problem of minimizing the spread of undesirable information by blocking a limited number of links in a social network [10], and to visualize a complex network in terms of information flow [19]. The IC model requires the parameters that represent diffusion probabilities through links to be specified in advance. However, the true values of the parameters are not available in practice. Thus, it is an important research issue to develop a method that can efficiently estimate them.

One of the drawbacks of the IC model is that it cannot represent time-delay for information propagation. Consider, as an example, modeling the

day-by-day spread of a topic in a blog network in which blog authors are connected to each other as is done in [7]. Here, a topic is a URL or phrase that can be tracked down from blog to blog. Suppose that there are blogroll links from two blog authors v and w to another blog author u. This means that v and w are readers of the blog of u. Suppose that both v and w publish posts on the topic that was addressed in u's post (meaning that v and w are both infected by u). There can be a difference between the dates that v and w publish their posts about the topic. Thus, Gruhl et al. [7] incorporated time-delay into the IC model. We refer to their model as the *ICTD model* [1]. Note that the ICTD model includes the IC model as a special case. The ICTD model is equipped with the parameters that represent time-delay through links as well as the parameters that represent diffusion probabilities through links.

They presented a method for estimating the values of these parameters from observed information diffusion results using an EM-like algorithm, and experimentally showed its effectiveness using sparse Erdös-Renyi networks. Here we note that large real social networks generally include dense subgraphs. For example, Newman and Park [16] observed that social networks represented as undirected graphs have high clustering coefficients, and positive correlations between the degrees of adjacent nodes. In these realistic networks it is important that the diffusion model explicitly addresses the possibility that a node is activated simultaneously by its multiple parent nodes each of which may have become activated at different time in the past. Their method [7], however, ignores this phenomenon and we speculate that their method does not perform well for dense networks, which is ex-

[1] It means the "IC with time-delay" model.

perimentally demonstrated in Section 6.2. In addition, it is not clear what they are optimizing in deriving the update formulas of the parameter values. We have already developed a method in [11] for estimating the values of the parameters in case of the IC model. The problem was much simpler because of no time-delay.

In this paper, we extend it to the ICTD model and propose a novel method for estimating the values of the parameters from a set of information diffusion results that are observed as time-sequences of infected (active) nodes. What makes this problem difficult is that incorporating time-delay makes the time-sequence observation data structural. There is no way of knowing from the data which node activated which other node that comes later in the sequence. Further, as the time progresses, the possible activation states increases exponentially, which also makes computation intractable. We introduce an objective function that represents the likelihood of obtaining the observed data sequences under the ICTD model on a given network, and derive an iterative algorithm by which the objective function is maximized. We experimentally show using both one synthetic and three real world networks that the proposed method outperforms the method by Gruhl et al. [7] for finding the correct parameters. We further show that it is crucially important that the parameters are estimated as accurately as possible in order to correctly predict the influence of nodes by which to rank and extract influential nodes.

Our contribution is that 1) we derived an algorithm in a principled way that guarantees convergence, avoids combinatorial explosion and can learn efficiently, from observed data, the parameters for the ICTD model, an information diffusion model that allows asynchronous time-delay and accounts

for multiple activation of a node, and 2) we showed that the algorithm performs satisfactorily for both synthetic and real social networks and outperforms the eGGLT method, a slightly modified version of [7], and a poor performance of the eGGLT method for a dense network is attributed to the ignorance of the multiple activation. We further point out that 3) the ranking method based on the proposed algorithm can be interpreted as a new concept of centrality based on information diffusion.

2 Information Diffusion Model and Learning Problem

We first recall the definition of the IC model according to [8], and then define the ICTD model by Gruhl et al. [7]. After that, we formulate our learning problem.

In this paper, we consider mathematical models for the spread of information through a directed network $G = (V, E)$ without self-links, where V and E ($\subset V \times V$) stands for the sets of all the nodes and links, respectively.

If there is a link (u, v) from node u to node v, node v is called a *child node* of node u and node u is called a *parent node* of node v. For each node $v \in V$, let $F(v)$ and $B(v)$ denote the set of child nodes of v and the set of parent nodes of v, respectively, i.e.,

$$\begin{aligned} F(v) &= \{w \in V;\ (v, w) \in E\}, \\ B(v) &= \{u \in V;\ (u, v) \in E\}. \end{aligned}$$

We call nodes *active* if they have been influenced with the information. In the information diffusion model, the diffusion process unfolds in a discrete time-step $t \geq 0$, and it is assumed that nodes can switch their states only

from inactive to active, but not from active to inactive. Given an initial active node v_0, we assume that the node v_0 has become active at time-step 0, and all the other nodes are inactive at time-step 0.

2.1 IC model

Here we define the IC model. In this model, for each link (u, v), we specify a real value $\kappa_{u,v}$ with $0 < \kappa_{u,v} < 1$ in advance, where $\kappa_{u,v}$ is referred to as the *diffusion probability* through link (u, v). The diffusion process proceeds from a given initial active node v_0 in the following way. When a node u becomes active at time-step t, it is given a single chance to activate each currently inactive child node v, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time-step $t + 1$. If multiple parent nodes of v become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

2.2 ICTD model

Gruhl et al. [7] extended the IC model so as to allow time-delay, and presented an information diffusion model with time-delay on network G. Now we call their model, the *ICTD model*.

In the ICTD model, for each link $(u, v) \in E$, we specify real values $r_{u,v}$ and $\kappa_{u,v}$ with

$$0 < r_{u,v},\, \kappa_{u,v} < 1$$

in advance. Gruhl et al. [7] considered modeling the spread of a topic in a

blog network, where blog authors are connected by a directed network. The parameter $r_{u,v}$ models how early author v reads the blog posts of author u, and the parameter $\kappa_{u,v}$ models the probability that author v, after reading the blog post of author u, publishes a blog post about the topic that author u addressed. Thus, $r_{u,v}$ and $\kappa_{u,v}$ were called the reading and the copy probabilities through link (u, v), respectively. In this paper, we refer to $r_{u,v}$ and $\kappa_{u,v}$ as the *time-delay parameter* and the *diffusion parameter* through link (u, v), respectively.

The diffusion process of the model proceeds from a given initial active node v_0 in the following way. Suppose that a node u becomes active at time-step t. Then, node u is given a single chance to activate each currently inactive child node v. We choose a delay-time δ from the geometric distribution with parameter $r_{u,v}$. If node v is not active at time-step $t + \delta$, then node u attempts to activate node v at time-step $t + \delta$, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time-step $t + \delta + 1$. If multiple parent nodes of v attempt to activate v at time-step $t + \delta$, then their activation attempts are sequenced in an arbitrary order. Whether or not u succeeds at time-step $t + \delta$, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

2.3 Learning problem

For the ICTD model on network G, we define the time-delay parameter vector $\boldsymbol{r}$ and the diffusion parameter vector $\boldsymbol{\kappa}$ by

$$\boldsymbol{r} = (r_{u,v})_{(u,v)\in E}, \quad \boldsymbol{\kappa} = (\kappa_{u,v})_{(u,v)\in E}.$$

In practice, the true values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ are not available. Thus, we must estimate them from past information diffusion histories observed as sets of active nodes.

We suppose that

$$\mathcal{D}_M = \{D_m; m = 1, \cdots, M\}$$

is an observed data set of M independent information diffusion results. Here, each D_m is a time-sequence of active nodes for the mth information diffusion result,

$$D_m = \langle D_m(0), D_m(1), \cdots, D_m(T_m)\rangle,$$

where $D_m(t)$ is the set of all the nodes that have first become active at time-step t, and $T_m(\geq 1)$ is the observed final time-step. We set

$$C_m(t) = D_m(0) \cup \cdots \cup D_m(t).$$

Note that $C_m(t)$ is the set of active nodes at time-step t for the mth information diffusion result. Note also that $C_m(T_m)$ is the set of all the active nodes for the mth information diffusion result. For any $v \in C_m(T_m)$, let $t_{m,v}$ denote the time-step at which node v becomes active for the mth information diffusion result, that is,

$$v \in D_m(t_{m,v}).$$

In this paper, we consider the problem of estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$.

3 Proposed Method

Here, we explain how we estimate the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$.

3.1 Likelihood function

For the learning problem described above, we derive the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ for use as our objective function in a rigorous manner.

First, we consider any node $v \in C_m(T_m)$ with $t_{m,v} > 0$ for the mth information diffusion result. Let $h_{m,v}$ denote the probability that the node v is activated at time $t_{m,v}$. We need to calculate $h_{m,v}$ in order to derive $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$. Here, to calculate $h_{m,v}$, we introduce an indicator vector

$$\boldsymbol{a}_{m,*,v} = (a_{m,u,v})_{u \in B(v) \cap C_m(t_{m,v}-1)},$$

where $a_{m,u,v} = 1$ if node u actually succeeded in activating v at time $t_{m,v}$; $a_{m,u,v} = 0$ otherwise. Note that if there exist multiple active parents for the node v, i.e.,

$$\eta = |B(v) \cap C_m(t_{m,v} - 1)| > 1,$$

it is not possible to know the exact values of $\boldsymbol{a}_{m,*,v}$ from the observed data. For example, in case of $B(v) \cap C_m(t_{m,v} - 1) = \{u, u'\}$, i.e., $\eta = 2$, we need to consider the following three possibilities;

$$\begin{aligned}
&1) \quad a_{m,u,v} = 1,\ a_{m,u',v} = 0,\\
&2) \quad a_{m,u,v} = 0,\ a_{m,u',v} = 1,\\
&3) \quad a_{m,u,v} = 1,\ a_{m,u',v} = 1.
\end{aligned}$$

Thus we can regard the indicator vector $\boldsymbol{a}_{m,*,v}$ as a latent vector, the number of hidden states of which amounts to as large as $2^\eta - 1$. This means that a naive calculation algorithm is likely to suffer from computational loads when η becomes larger. To cope with this difficulty, we develop efficient computation formulas (see Equations (5) and (16) below).

Let $\mathcal{A}_{m,u,v}$ denote the probability that a node $u \in B(v) \cap C_m(t_{m,v}-1)$ activates the node v at time $t_{m,v}$, that is,

$$\mathcal{A}_{m,u,v} = \kappa_{u,v} r_{u,v} (1-r_{u,v})^{t_{m,v}-t_{m,u}-1}. \tag{1}$$

Let $\mathcal{B}_{m,u,v}$ denote the probability that the node v is not activated from a node $u \in B(v) \cap C_m(t_{m,v}-1)$ within the time-period $[t_{m,u}+1, t_{m,v}]$, that is,

$$\begin{aligned} \mathcal{B}_{m,u,v} &= 1 - \kappa_{u,v} \sum_{t=t_{m,u}+1}^{t_{m,v}} r_{u,v}(1-r_{u,v})^{t-t_{m,u}-1} \\ &= \kappa_{u,v}(1-r_{u,v})^{t_{m,v}-t_{m,u}} + (1-\kappa_{u,v}). \end{aligned} \tag{2}$$

By using the indicator vector $\boldsymbol{a}_{m,*,v}$, the probability $h_{m,v}$ can naturally be expressed as

$$h_{m,v} = \sum_{\boldsymbol{a}_{m,*,v} \neq \mathbf{0}} f_{m,v}(\boldsymbol{a}_{m,*,v}), \tag{3}$$

where the summation is taken over all non-zero indicator (binary) vectors, and

$$f_{m,v}(\boldsymbol{a}_{m,*,v}) = \prod_{u \in B(v) \cap C_m(t_{m,v}-1)} (\mathcal{A}_{m,u,v})^{a_{m,u,v}} (\mathcal{B}_{m,u,v})^{1-a_{m,u,v}}. \tag{4}$$

Note that $f_{m,v}(\boldsymbol{a}_{m,*,v})$ is the probability that node v is activated at time $t_{m,v}$ according to the indicator vector $\boldsymbol{a}_{m,*,v}$. In order to efficiently calculate $h_{m,v}$, we consider the following transformation:

$$h_{m,v} = \sum_{\boldsymbol{a}_{m,*,v}} f_{m,v}(\boldsymbol{a}_{m,*,v}) - \prod_{u \in B(v) \cap C_m(t_{m,v}-1)} \mathcal{B}_{m,u,v},$$

where the summation is taken over all indicator (binary) vectors. Thus, by Equation (4), we have

$$h_{m,v} = \prod_{u \in B(v) \cap C_m(t_{m,v}-1)} (\mathcal{A}_{m,u,v} + \mathcal{B}_{m,u,v}) \; - \prod_{u \in B(v) \cap C_m(t_{m,v}-1)} \mathcal{B}_{m,u,v}. \tag{5}$$

Therefore, by using Equation (5), we can calculate $h_{m,v}$ efficiently without considering all the possible occurrences of non-zero indicator vectors.

Next, for the mth information diffusion result, we consider any link $(v, w) \in E$ such that $v \in C_m(T_m)$ and $w \notin C_m(T_m)$. Let $g_{m,v,w}$ denote the probability that the node w is not activated by the node v within the observed time-period $[0, T_m]$. We can easily derive the following equation:

$$g_{m,v,w} = \kappa_{v,w}(1 - r_{v,w})^{T_m - t_{m,v}} + (1 - \kappa_{v,w}). \tag{6}$$

Here we can naturally assume that each information diffusion process finished sufficiently earlier than the observed final time, i.e.,

$$T_m \gg \max\{t; D_m(t) \neq \emptyset\}.$$

Thus, as $T_m \to \infty$ in Equation (6), we assume

$$g_{m,v,w} \ = \ 1 - \kappa_{v,w}. \tag{7}$$

Therefore, by using Equations (5) and (7), and the independence properties, we can define the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ by

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M) = \prod_{m=1}^{M} \left(\prod_{t=1}^{T_m} \prod_{v \in D_m(t)} h_{m,v} \right) \left(\prod_{v \in C_m(T_m)} \prod_{w \in F(v) \setminus C_m(T_m)} g_{m,v,w} \right), \tag{8}$$

where $F(v) \setminus C_m(T_m)$ denotes the set difference of $F(v)$ and $C_m(T_m)$, that is, $F(v) \setminus C_m(T_m) = \{w \in F(v);\ w \notin C_m(T_m)\}$. In this paper, we focus on the above situation (i.e., Equation (7)) for simplicity, but we can easily modify our method to cope with the general one (i.e., Equation (6)). Thus, our problem is to obtain the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, which maximize Equation (8). For

this estimation problem, we derive a method based on an iterative algorithm in order to stably obtain its solution.

3.2 Estimation method

We describe our method for estimating the optimal values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. We suppose that

$$\overline{\boldsymbol{r}} = (\overline{r}_{u,v})_{(u,v)\in E}, \quad \overline{\boldsymbol{\kappa}} = (\overline{\kappa}_{u,v})_{(u,v)\in E}$$

are the current estimates of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, respectively. We derive update formulas of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ such that

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M) \geq \mathcal{L}(\overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}; \mathcal{D}_M).$$

For each $v \in C_m(T_m)$ and indicator vector $\boldsymbol{a}_{m,*,v}$, we define $q_{m,v}(\boldsymbol{a}_{m,*,v})$ by

$$q_{m,v}(\boldsymbol{a}_{m,*,v}) = \frac{f_{m,v}(\boldsymbol{a}_{m,*,v})}{h_{m,v}}. \tag{9}$$

Note that $q_{m,v}(\boldsymbol{a}_{m,*,v})$ is the posterior probability that the indicator vector is $\boldsymbol{a}_{m,*,v}$ when v is activated at time $t_{m,v}$ (see Equations (3), (4)). Let $\overline{\mathcal{A}}_{m,u,v}$, $\overline{\mathcal{B}}_{m,u,v}$, $\overline{h}_{m,v}$, and $\overline{q}_{m,v}(\boldsymbol{a}_{m,*,v})$ denote the values of $\mathcal{A}_{m,u,v}$, $\mathcal{B}_{m,u,v}$, $h_{m,v}$, and $q_{m,v}(\boldsymbol{a}_{m,*,v})$ calculated by using $\overline{\boldsymbol{r}}$ and $\overline{\boldsymbol{\kappa}}$, respectively.

From Equations (3), (7), and (8), we can transform our objective function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ as follows:

$$\log \mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M) = Q(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}) - H(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}). \tag{10}$$

Here, $Q(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ is defined by

$$Q(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}) = \tag{11}$$
$$\sum_{m=1}^{M} \left(\sum_{t=1}^{T_m} \sum_{v \in D_m(t)} Q_{m,v} + \sum_{v \in C_m(T_m)} \sum_{w \in F(v) \setminus C_m(T_m)} \log(1 - \kappa_{v,w}) \right),$$

where

$$Q_{m,v} = \sum_{\boldsymbol{a}_{m,*,v} \neq \mathbf{0}} \overline{q}_{m,v}(\boldsymbol{a}_{m,*,v}) \log f_{m,v}(\boldsymbol{a}_{m,*,v}). \tag{12}$$

Also, $H(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ is defined by

$$H(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}) = \sum_{m=1}^{M} \sum_{t=1}^{T_m} \sum_{v \in D_m(t)} \sum_{\boldsymbol{a}_{m,*,v} \neq \mathbf{0}} J_{m,v}(\boldsymbol{a}_{m,*,v}), \tag{13}$$

where

$$J_{m,v}(\boldsymbol{a}_{m,*,v}) = \overline{q}_{m,v}(\boldsymbol{a}_{m,*,v}) \log q_{m,v}(\boldsymbol{a}_{m,*,v}). \tag{14}$$

Since the function $H(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ is maximized at $\boldsymbol{r} = \overline{\boldsymbol{r}}$ and $\boldsymbol{\kappa} = \overline{\boldsymbol{\kappa}}$ from Equations (13) and (14), we can increase the value of $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ by maximizing the function $Q(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ (see Equation (10)).

In order to efficiently calculate $Q_{m,v}$, we derive the following formula from Equations (4), (9) and (12):

$$Q_{m,v} = \sum_{u \in B(v) \cap C_m(t_{m,v}-1)} \left(\overline{\alpha}_{m,u,v} \log \mathcal{A}_{m,u,v} + (1 - \overline{\alpha}_{m,u,v}) \log \mathcal{B}_{m,u,v} \right), \tag{15}$$

where

$$\overline{\alpha}_{m,u,v} = \sum_{\boldsymbol{a}_{m,*,v} \neq \mathbf{0}} a_{m,u,v}\, \overline{q}_{m,v}(\boldsymbol{a}_{m,*,v}).$$

Note that $0 < \overline{\alpha}_{m,u,v} < 1$. Then, we can easily show that

$$\overline{\alpha}_{m,u,v} = \frac{\overline{\mathcal{A}}_{m,u,v}}{\overline{h}_{m,v}} \prod_{x \in B(v) \cap C_m(t_{m,v}-1) \setminus \{u\}} \left(\overline{\mathcal{A}}_{m,x,v} + \overline{\mathcal{B}}_{m,x,v} \right). \tag{16}$$

Therefore, by using Equation (16), we can also calculate $\overline{\alpha}_{m,u,v}$ without computation of exponential order.

Note here that although $\log \mathcal{A}_{m,u,v}$ is a linear combination of $\log r_{u,v}$, $\log(1 - r_{u,v})$, and $\log \kappa_{u,v}$, $\log \mathcal{B}_{m,u,v}$ cannot be written by such a linear combination (see Equations (1) and (2)). In order to cope with this problem of $\log \mathcal{B}_{m,u,v}$, we transform $\log \mathcal{B}_{m,u,v}$ in the same way as Equation (10):

$$\log \mathcal{B}_{m,u,v} = Q^{\mathcal{B}}_{m,u,v} - H^{\mathcal{B}}_{m,u,v}, \tag{17}$$

where

$$Q^{\mathcal{B}}_{m,u,v} = \overline{\beta}_{m,u,v} \log \left(\kappa_{u,v} (1 - r_{u,v})^{t_{m,v} - t_{m,u}} \right) + (1 - \overline{\beta}_{m,u,v}) \log(1 - \kappa_{u,v}), \quad (18)$$

and

$$H^{\mathcal{B}}_{m,u,v} = \overline{\beta}_{m,u,v} \log \beta_{m,u,v} + (1 - \overline{\beta}_{m,u,v}) \log(1 - \beta_{m,u,v}). \quad (19)$$

Here, $\beta_{m,u,v}$ is defined by

$$\beta_{m,u,v} = \frac{\kappa_{u,v}(1 - r_{u,v})^{t_{m,v} - t_{m,u}}}{\kappa_{u,v}(1 - r_{u,v})^{t_{m,v} - t_{m,u}} + (1 - \kappa_{u,v})}, \quad (20)$$

and $\overline{\beta}_{m,u,v}$ is the value of $\beta_{m,u,v}$ calculated by using $\overline{\boldsymbol{r}}$ and $\overline{\boldsymbol{\kappa}}$. Note that $0 < \overline{\beta}_{m,u,v} < 1$. We define $Q'(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ by

$$Q'(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}) = \sum_{m=1}^{M} \left(\sum_{t=1}^{T_m} \sum_{v \in D_m(t)} {Q'}_{m,v} + \sum_{v \in C_m(T_m)} \sum_{w \in F(v) \setminus C_m(T_m)} \log(1 - \kappa_{v,w}) \right), \quad (21)$$

where

$${Q'}_{m,v} = \sum_{u \in B(v) \cap C_m(t_{m,v} - 1)} \left(\overline{\alpha}_{m,u,v} \log \mathcal{A}_{m,u,v} + (1 - \overline{\alpha}_{m,u,v}) Q^{\mathcal{B}}_{m,u,v} \right). \quad (22)$$

Note that by Equations (19) and (20), the function $H^{\mathcal{B}}_{m,u,v}$ of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ is maximized at $\boldsymbol{r} = \overline{\boldsymbol{r}}$ and $\boldsymbol{\kappa} = \overline{\boldsymbol{\kappa}}$. Thus, we can maximize $Q(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ by maximizing $Q'(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ as functions of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, (see Equations (12), (15), (17), (21) and (22)). Note here that the function $Q'(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ is a linear combination of $\{\log r_{u,v}$, $\log(1 - r_{u,v})$, $\log \kappa_{u,v}$, $\log(1 - \kappa_{u,v})$; $(u, v) \in E\}$ with positive coefficients.

From Equations (1), (18), (21) and (22), we can easily obtain the solution which maximizes $Q'(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$. Hence, we have the following theorem.

Theorem 1 *Let* $\overline{\boldsymbol{r}} = (\overline{r}_{u,v})$ *and* $\overline{\boldsymbol{\kappa}} = (\overline{\kappa}_{u,v})$ *be the current estimates of* $\boldsymbol{r}$ *and* $\boldsymbol{\kappa}$*, respectively. For each* $(u, v) \in E$ *and* $m \in \{1, \cdots, M\}$*, we define* $\mathcal{M}^+_{u,v}$*,* $\mathcal{M}^-_{u,v}$*, and* $\overline{\varphi}_{m,u,v}$ *by*

$$\mathcal{M}^+_{u,v} = \{m \in \{1, \cdots, M\};\ u, v \in C_m(T_m),\ v \in F(u),\ t_{m,u} < t_{m,v}\}, \quad (23)$$

$$\mathcal{M}^-_{u,v} = \{m \in \{1, \cdots, M\};\ u \in C_m(T_m),\ v \notin C_m(T_m),\ v \in F(u)\}, \quad (24)$$

and

$$\overline{\varphi}_{m,u,v} = \overline{\alpha}_{m,u,v} + (1 - \overline{\alpha}_{m,u,v})\ \overline{\beta}_{m,u,v},$$

respectively. Then, if we update the values of $\boldsymbol{r}$ *and* $\boldsymbol{\kappa}$ *by*

$$r_{u,v} = \frac{\sum_{m \in \mathcal{M}^+_{u,v}} \overline{\alpha}_{m,u,v}}{\sum_{m \in \mathcal{M}^+_{u,v}} (t_{m,v} - t_{m,u})\ \overline{\varphi}_{m,u,v}}, \quad (25)$$

$$\kappa_{u,v} = \frac{\sum_{m \in \mathcal{M}^+_{u,v}} \overline{\varphi}_{m,u,v}}{|\mathcal{M}^+_{u,v}| + |\mathcal{M}^-_{u,v}|} \quad (26)$$

for all $(u, v) \in E$*, we have*

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M) \geq \mathcal{L}(\overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}; \mathcal{D}_M).$$

Theorem 1 provides the update formulas (25) and (26) of the parameters $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. Since the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ is ensured not to decrease by this updating, this updating mechanism asymptotically converges to at least locally optimal solutions, and we can propose a method similar to EM algorithm in a natural way. Note that each $h_{m,u,v}$ and $\overline{\alpha}_{m,u,v}$ are efficiently calculated by Equations (5) and (16), respectively.

4 Evaluation Methods

We first evaluate the estimation accuracy of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ for the proposed learning method. Next, we exploit the estimated model to predict the influence of nodes and extract the influential nodes, and evaluate their performance.

For an initial active node v, let $\psi(v; \boldsymbol{r}, \boldsymbol{\kappa})$ denote the number of active nodes at the end of the information diffusion process for the ICTD model with parameter values $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. Note that $\psi(v; \boldsymbol{r}, \boldsymbol{\kappa})$ is a random variable since the information diffusion process is a random process. Let $\sigma(v; \boldsymbol{r}, \boldsymbol{\kappa})$ denote the expected value of $\psi(v; \boldsymbol{r}, \boldsymbol{\kappa})$. We call $\sigma(v; \boldsymbol{r}, \boldsymbol{\kappa})$ the *influence degree* of node v for the ICTD model with parameter values $\boldsymbol{r}$ and $\boldsymbol{\kappa}$.

When we are given the set of information diffusion results $\mathcal{D}_M$, we measure the influence of node v by the influence degree $\sigma(v; \boldsymbol{r}^*, \boldsymbol{\kappa}^*)$ for the ICTD model which generated $\mathcal{D}_M$, where $\boldsymbol{r}^*$ and $\boldsymbol{\kappa}^*$ are the true values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, respectively. Thus, a node v with high influence degree $\sigma(v; \boldsymbol{r}^*, \boldsymbol{\kappa}^*)$ is an influential node. We first estimate the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$. Suppose that $\widehat{\boldsymbol{r}}$ and $\widehat{\boldsymbol{\kappa}}$ are the estimated values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, respectively. Then, we predict $\sigma(v; \widehat{\boldsymbol{r}}, \widehat{\boldsymbol{\kappa}})$ and use it as the estimated value of the influence degree $\sigma(v; \boldsymbol{r}^*, \boldsymbol{\kappa}^*)$ of node v. Moreover, we extract the influential nodes by ranking nodes v based on $\sigma(v; \widehat{\boldsymbol{r}}, \widehat{\boldsymbol{\kappa}})$.

We evaluate the proposed method in terms of the capability of both predicting the influence degrees of nodes and ranking the influential nodes. We focus on the performance for high-rank nodes since we are interested in influential nodes. Let $L^*(k)$ denote the set of top k influential nodes for the true ICTD model. Let $\widehat{L}(k)$ be the set of top k influential nodes estimated by a given ranking method. We evaluate the performance of the ranking method by the *ranking similarity* $\mathcal{F}(k)$ within the rank k, where $\mathcal{F}(k)$ is defined by

$$\mathcal{F}(k) = \frac{|L^*(k) \cap \widehat{L}(k)|}{k}. \tag{27}$$

5 Comparison Methods

5.1 eGGLT method

Gruhl et al. [7] presented a method for estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$ for the ICTD model, from which the underlying link structure E was induced. Thus, their method did not explicitly use the link structure E, and $\boldsymbol{r} = (r_{u,v})$ and $\boldsymbol{\kappa} = (\kappa_{u,v})$ were first regarded as full $|V|(|V|-1)$ dimensional vectors. In practice, by exploiting some heuristics based on the observed data $\mathcal{D}_M$, they restricted the non-zero entries of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ in order to achieve a reduction in computational cost, and inferred the underlying link structure from the estimated values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ under these constraints. However, their method can be straightforwardly extended to the case where the underlying network structure is known. We refer to the extended method as the *eGGLT method* [2], and compare the proposed method with the eGGLT method.

We begin with describing the original method by Gruhl et al. [7]. Basically, they presented an EM-like algorithm. Let $\overline{\boldsymbol{r}} = (\overline{r}_{u,v})$ and $\overline{\boldsymbol{\kappa}} = (\overline{\kappa}_{u,v})$ be the current estimates of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, respectively. The update formulas for $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ are as follows:

$$r_{u,v} = \frac{\sum_{m \in \mathcal{S}^+_{u,v}} \overline{p}_{m,u,v}}{\sum_{m \in \mathcal{S}^+_{u,v}} (t_{m,v} - t_{m,u})\ \overline{p}_{m,u,v}}, \tag{28}$$

$$\kappa_{u,v} = \frac{\sum_{m \in \mathcal{S}^+_{u,v}} \overline{p}_{m,u,v}}{\sum_{m \in \mathcal{S}^+_{u,v}} \overline{\lambda}_{m,u,v} + \sum_{m \in \mathcal{S}^-_{u,v}} \overline{\mu}_{m,u,v}}, \tag{29}$$

for all $u, v \in V$ with $u \neq v$, where

$$\overline{p}_{m,u,v} = \frac{\overline{r}_{u,v}(1-\overline{r}_{u,v})^{t_{m,v}-t_{m,u}-1}\overline{\kappa}_{u,v}}{\sum_{w \in C_m(t_{m,v}-1)} \overline{r}_{w,v}(1-\overline{r}_{w,v})^{t_{m,v}-t_{m,w}-1}\overline{\kappa}_{w,v}}, \tag{30}$$

and

$$\overline{\lambda}_{m,u,v} = 1 - (1-\overline{r}_{u,v})^{t_{m,v}-t_{m,u}},$$

[2] It means the extended "Gruhl, Guha, Liben-Nowell, and Tomkins [7]" method.

$$\overline{\mu}_{m,u,v} \quad = \quad 1 - (1 - \overline{r}_{u,v})^{T_m - t_{m,u}}.$$

Here, $\mathcal{S}^+_{u,v}$ and $\mathcal{S}^-_{u,v}$ are defined by

$$\mathcal{S}^+_{u,v} \quad = \quad \{m \in \{1, \cdots, M\};\ u, v \in C_m(T_m),\ t_{m,u} < t_{m,v}\}, \tag{31}$$

$$\mathcal{S}^-_{u,v} \quad = \quad \{m \in \{1, \cdots, M\};\ u \in C_m(T_m),\ v \notin C_m(T_m)\}. \tag{32}$$

We note that they did not derive the EM-like algorithm in a principled way. No objective function is defined to obtain the updating formulas. Here we should also mention that this method might have an intrinsic limitation because simultaneous activations from multiple parent nodes are not considered in Equation (30).

Now, we define the eGGLT method. The eGGLT method straightforwardly incorporates the link structure E into the original method by Gruhl et al. [7]. Namely, the eGGLT method only changes the update formulas (28) and (29) of $r_{u,v}$ and $\kappa_{u,v}$ for any link $(u, v) \in E$ as follows: $\mathcal{S}^+_{u,v}$ and $\mathcal{S}^-_{u,v}$ are replaced with $\mathcal{M}^+_{u,v}$ and $\mathcal{M}^-_{u,v}$, respectively (see Equations (31), (32), (23) and (24)). Note here that if the underlying network $G = (V, E)$ is a complete graph, we have $\mathcal{S}^+_{u,v} = \mathcal{M}^+_{u,v}$ and $\mathcal{S}^-_{u,v} = \mathcal{M}^-_{u,v}$ for all $(u, v) \in E$. The eGGLT method can be applied to the problem of estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$ when the underlying network structure is known. Therefore, the eGGLT method can be also applied to the problems of predicting the influence degrees of nodes for the true ICTD model and ranking the influential nodes.

5.2 Conventional methods in social network analysis

As for the problem of extracting the high ranked influential nodes for the true ICTD model, we also compare the proposed method with four heuristics

from social network analysis, which are the degree, the betweenness, the closeness, and the PageRank methods.

First, "degree centrality", "betweenness centrality", and "closeness centrality" are commonly used as influence measure for a bidirectional network in sociology [20], where the degree of node v is defined as the number of links attached to v, the betweenness of node v is defined as the total number of shortest paths between pairs of nodes that pass through v, and the closeness of node v is defined as the reciprocal of the average distance between v and other nodes in the network.

We also consider measuring the influence of each node by its "authoritativeness" obtained by the "PageRank" method [3], since this is a well known method for identifying authoritative or influential pages in a hyperlink network of web pages. This method has a parameter ε; when we view it as a model of a random web surfer, ε corresponds to the probability with which a surfer jumps to a page picked uniformly at random [17]. In our experiments, we used a typical setting of $\varepsilon = 0.15$.

6 Experimental Evaluation

We first compared the proposed method with the eGGLT method for the capability of estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ in the case of a complete graph. Next, using three large real social networks, we evaluated the effectiveness of the proposed method for 1) estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, 2) predicting the influence degrees of the high-ranked nodes under the true ICTD model, and 3) ranking the influential nodes.

6.1 Experimental settings

According to [7], we generated the training data $\mathcal{D}_M$ by simulating the true ICTD model M_0 times from every single node as being an initial active node, where $M = M_0|V|$. When we estimate the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$, we always used the same initial guess and the same iteration number for the proposed and the eGGLT methods. Strictly speaking, we always set the initial values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ as $r_{u,v} = 1/2$ and $\kappa_{u,v} = 1/2$ for any $(u, v) \in E$, and performed 100 iterations. We confirmed that the relative difference of the parameter values in the successive iteration is in the order of 10^{-5}, and thus, the solutions are judged to be converged.

We note that the influence degree $\sigma(v; \boldsymbol{r}, \boldsymbol{\kappa})$ of a node v is invariant with respect to the values of the delay-parameters $\boldsymbol{r}$. Thus, we can calculate the $\sigma(v; \boldsymbol{r}, \boldsymbol{\kappa})$ of the ICTD model by the influence degree of v for the corresponding IC model. Hence, we evaluated the influence degrees $\{\sigma(v; \boldsymbol{r}, \boldsymbol{\kappa}); v \in V\}$ by applying the method of [9] with the parameter value $10,000$ to the corresponding IC model, where the parameter represents the number of bond percolation processes (see [9] for more details).

6.2 Synthetic network

We recall that the proposed method considers the possibility that a node v can be activated simultaneously by multiple parent nodes $\{u\}$ that has become activated at different times, whereas the eGGLT method does not assume this possibility. Thus, we first consider a network with the highest clustering coefficients (see [15]), where there is a large possibility that the above situations happen. Here, we exploited the complete directed graph of 50 nodes as the network $G = (V, E)$. Note that the eGGLT method

coincides with the original method by Gruhl et al. [7] for a complete graph. According to [7], we used $r_{u,v} = 2/3$ and $\kappa_{u,v} = 1/10$ for any $(u, v) \in E$ as the true values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. We refer to this dataset as the complete graph dataset.

We compared the capability of estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ for the proposed and the eGGLT methods. Table 1 shows the results for different number of simulations M_0 for all nodes. Here, mean($\boldsymbol{r}$) and mean($\boldsymbol{\kappa}$) denote the means of the estimated values of $\boldsymbol{r} = (r_{u,v})$ and $\boldsymbol{\kappa} = (\kappa_{u,v})$, respectively, and std($\boldsymbol{r}$) and std($\boldsymbol{\kappa}$) denote their standard deviations, respectively. The results demonstrate that the proposed method outperforms the eGGLT method. Our algorithm can converge to the true values efficiently when there is a reasonable amount of training data. The results demonstrate the effectiveness of the proposed method.

6.3 Real networks

6.3.1 Network dataset

As a large real social network $G = (V, E)$, we first employed the blog network used in [10]. This was a bidirectional network with $12,047$ nodes and $79,920$ directed links. Again, we used $r_{u,v} = 2/3$ and $\kappa_{u,v} = 1/10$ for any $(u, v) \in E$ as the true values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. We refer to this dataset as the blog network dataset.

Second, we employed a network derived from the Enron Email Dataset [12]. We first extracted the email addresses that appeared in the Enron Email Dataset as senders and recipients. We regarded each email address as a node, and constructed an undirected network obtained by linking two email

addresses u and v if u sent an email to v and received an email from v. Next, we extracted its maximal strongly connected component, and constructed a directed network by regarding those undirected links as bidirectional ones. We refer to this strongly connected bidirectional network as the Enron network. This network had $4,254$ nodes and $44,314$ directed links. Again, we used $r_{u,v} = 2/3$ and $\kappa_{u,v} = 1/10$ for any $(u, v) \in E$ as the true values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. We refer to this dataset as the Enron network dataset.

Third, we employed the co-authorship network used in [18]. This was a bidirectional network with $12,357$ nodes and $38,896$ directed links. For the co-authorship network, we used $r_{u,v} = 2/3$ and $\kappa_{u,v} = 1/5$ for any $(u, v) \in E$ as the true values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, since the mean out-degree of the co-authorship network is much smaller than the blog and the Enron networks. In fact, when we used $\kappa_{u,v} = 1/10$ for any link $(u, v) \in E$, the influence degrees of nodes became very low (they were less than 15). We refer to this dataset as the co-authorship network dataset.

6.3.2 Experimental results

First, we evaluated the performance for estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. Tables 2, 3 and 4 show the results for different number of simulations M_0 for all nodes in the blog, the Enron and the co-authorship network datasets, respectively. Here, the meanings of mean($\boldsymbol{r}$), std($\boldsymbol{r}$), mean($\boldsymbol{\kappa}$), and std($\boldsymbol{\kappa}$) are the same as in Table 1. We observe that the proposed method outperforms the eGGLT method for all of these three datasets in estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, and can converge to the true values efficiently when there is a reasonable amount of training data.

Next, we investigated the performance for predicting the influence de-

grees of the top 200 influential nodes for the true ICTD model. As described in Section 4, we predicted $\sigma(v_k^*; \widehat{\boldsymbol{r}}, \widehat{\boldsymbol{\kappa}})$ and used it as the influence degree $\sigma(v_k^*; \boldsymbol{r}^*, \boldsymbol{\kappa}^*)$ of node v_k^* $(k = 1, \cdots, 200)$, where v_k^* denotes the node of rank k for the true ICTD model. Figs. 1, 2 and 3 show the results in the case of $M_0 = 60$ for the blog, the Enron and the co-authorship network datasets, respectively. In each figure, the thick dashed line displays the true influence degrees, and circles and squares indicate the influence degrees predicted by the proposed and the eGGLT methods, respectively. We observe from these figures that unlike the eGGLT method, the proposed method makes good predictions of the true influence degrees of the high-ranked nodes for all of these three datasets. Here, we evaluated the *mean influence-prediction error* $\mathcal{E}(k)$ within the top rank k, which is defined by

$$\mathcal{E}(k) = \frac{1}{k} \sum_{i=1}^{k} |\sigma(v_i^*; \boldsymbol{r}^*, \boldsymbol{\kappa}^*) - \sigma(v_i^*; \widehat{\boldsymbol{r}}, \widehat{\boldsymbol{\kappa}})|,$$

and also evaluated the mean $\mathcal{H}(k)$ of the true influence degrees $\{\sigma(v_i^*; \boldsymbol{r}^*, \boldsymbol{\kappa}^*);$ $i = 1, \cdots, k\}$ for the top k nodes. Table 5 compares the errors by the two methods for the three datasets in case of $k = 20$ and $k = 200$. We see that the mean influence-prediction error $\mathcal{E}(k)$ of the proposed method is much smaller than 1% of the mean of the true influence degrees $\mathcal{H}(k)$ regardless of the value of k for every dataset, whereas, in most cases, $\mathcal{E}(k)$ of the eGGLT method is more than 50 times greater than $\mathcal{E}(k)$ of the proposed method and more than 10% of $\mathcal{H}(k)$. Even in the best case where the percentage error of the eGGLT method is minimum, i.e., 6.6% of $\mathcal{H}(k)$ for $k = 20$ for the Enron network dataset, $\mathcal{E}(k)$ of the eGGLT method is about 500 times greater than $\mathcal{E}(k)$ of the proposed method. Therefore, we see that for all of these three datasets the proposed method works well and gives far better results than the eGGLT method.

Next, we evaluated the performance for ranking the top 200 influential nodes under the true ICTD model. Figs. 4, 5 and 6 show the ranking similarity $\mathcal{F}(k)$ (see Equation (27)) as a function of the top rank k in the case of $M_0 = 60$ for the blog, the Enron and the co-authorship network datasets, respectively. Here, circles, squares, triangles, diamonds, crosses, and asterisks indicate the results for the proposed, the eGGLT, the degree, the betweenness, the closeness, and the PageRank methods, respectively. For the blog network dataset, the proposed method performed best, and the eGGLT method followed. The other methods (the conventional methods in social network analysis) were much worse than these two methods. For the Enron network dataset, the proposed method performed best, and the eGGLT and the out-degree methods followed. The performance difference between the eGGLT and the out-degree methods was small. For example, the eGGLT method was worse than the out-degree method for the ranking similarity within rank $k = 200$. For the co-authorship network dataset, the proposed method performed best, and the eGGLT method followed. The other methods (the conventional methods in social network analysis) were much worse than these two methods. Therefore, we see for these datasets that the proposed method works effectively.

7 Discussion

We note that the results of the eGGLT method were not good in Table 1, contrary to the results in [7]. We attribute this to the inadequacy of the parameter estimation methods and the different setting of network parameters. First, as stated earlier, note that the proposed method considers the

possibility that a node v can be activated simultaneously by multiple parents $\{u\}$ that has become activated at different times (although v is activated only once), whereas the eGGLT method does not assume this possibility. Second note that the networks used in [7] are modified Erdös-Renyi random graphs with $|V| = 1000$ and d (the degree of node) $= 3$. This gives a sparse graph with almost 0 clustering coefficients (see [15]). Further, the diffusion parameter $\kappa_{u,v}$ they used for any link (u, v) is $1/10$, which implies that the above multiple activation possibility is essentially 0. However, the network we used in our experiment in Section 6.2 is a complete graph with very high clustering coefficients (i.e., clustering coefficient 1), and there is a large possibility that the above situations happen. In this setting, ignoring the possibility of a node being activated simultaneously from more than one parent node would most probably give inaccurate estimates of the parameters. The results in Section 6.2 are consistent with this observation. For the large real social networks used in our experiments, the mean clustering coefficients of the blog, the Enron and the co-authorship networks were 0.262, 0.370 and 0.218, respectively. This means that the Enron network has a larger possibility that the above situations happen than the blog and the co-authorship networks. Tables 2, 3 and 4 show that the parameter estimation results of the eGGLT method for the Enron network were worst. This is consistent with the discussion above.

Compared with the eGGLT method, our method derives the learning algorithm in a principled way. It has the objective function which has a clear meaning of the likelihood of obtaining the observed data, and the parameter updating algorithm is derived such that it iteratively increases the likelihood with the convergence guaranteed. Therefore, for large real social

networks, the proposed method far outperformed the eGGLT method for predicting influence degrees of true high-ranked nodes (see Figs. 1, 2 and 3), and always gave better results than the eGGLT method for ranking the influential nodes (see Figs. 4, 5 and 6). These results also imply that estimating the parameters as accurately as possible is very important. Further note that, in deriving the proposed algorithm, tactics are employed to avoid computational explosion.

We consider that our ranking method presents a novel concept of centrality based on the information diffusion model, i.e., the ICTD model. Actually, Figs. 4, 5 and 6 show that nodes identified as higher ranked by our method are substantially different from those by each of the conventional methods in social network analysis. This means that our method enables a new type of social network analysis if past information diffusion data are available. Of course, it is beyond controversy that each conventional method has its own merit and usage, and our method is an addition to them which has a different merit in terms of information diffusion.

The formulation we showed in Section 3.2 dealt with the case where each of $r_{u,v}$ and $\kappa_{u,v}$ can take a different value for each link $(u, v) \in E$. However, this would cause a serious problem of overfitting as well as unacceptably high computation cost if we are to analyze large real networks with high clustering coefficients. Parameter sharing helps improve generalization capability. Further, it is more realistic that some of the parameters share the same values across different links. In our framework, placing constraints, e.g., assigning uniform values to parameters across all links or grouping the parameters $(\kappa_{u,v})$ and $(r_{u,v})$ into several categories, etc. is straightforward. For example, we can divide the link set E into subsets $\{E_1, E_2, \cdots, E_N\}$

and assign unique parameter values r_n and κ_n for all the links in each category E_n. If the overfitting is the real problem even after grouping links, we can follow the standard approach of introducing prior distributions over the model parameters. For placing constraints in a more realistic setting, we may divide the nodes into two groups: those that strongly influence others and those not, or we may divide the nodes into another two groups: those that are easily influenced by others and those not. We can further divide the nodes into multiple groups. If there is some background knowledge about the node grouping, our method can make the best use of it, one of the characteristics of the artificial intelligence approach. Obtaining such background knowledge is also an important research topic in the knowledge discovery from social networks.

The final objective of Gruhl et al.'s work [7] was to estimate network structure. We did not focus on this aspect in this paper, however. This does not mean that our proposed method is unable to estimate the structure. Just like Gruhl et al. assumed a fully connected complete graph, we could have taken the same approach, i.e., initially assuming the complete network and deleting links for which no parameter values are obtained or the values are very small under the assumption that the observed sequence data is sufficiently large enough to cover all the possible information propagation paths. However, scalability becomes an issue with this naive approach. It is too computationally expensive to be applied for a real network, e.g. for a network with 10,000 nodes, the number of links, thus the number of each parameter, to be considered is 100,000,000. With such a huge number of parameters to search, both methods become infeasible. In addition, the amount of observation data that is required to estimate the parame-

ter values is tremendously large and it is almost unrealistic to collect such data. Better approaches including parameter sharing mentioned above must be yet explored so as to pose strong constraints on network structure and reasonably and effectively restrict parameters to be considered. This is our future work.

Information diffusion with time-delay we picked in this paper poses an interesting machine learning problem. Each piece of training data is a sequence of observed data (thus, relational), but it has some hidden structure and it is not straightforward to map the data to node-to-node information diffusion. In theory we have to consider all the possible paths to each activated node from unknown source nodes with different time-delays. We managed this by introducing indicator variables. How to avoid computational explosion then became crucial and we introduced neat tactics.

Our method utilizes sequential data of information diffusion. In this aspect, it has some commonality with re-enforcement learning, but the main difference is that a reward for each sequence is not used in our learning framework. Our model is meant to be useful for analyzing information diffusion via a human network, e.g., via words-of-mouth. It is not clear at the moment whether the similar approach can be used for analyzing a diffusion process in other domains, e.g., biological networks. If a similar model is confirmed to be usable, our method can also be an important technique to analyze general diffusion process. We plan to apply the proposed method to some specific tasks in a more practical setting, in which case the evaluation must be based on a task-specific performance measure for each task.

8 Conclusion

We addressed the problem of estimating the parameters for an information diffusion model (i.e., the ICTD model) in a complex social network, given the network topology and the observed time-sequence data. The model allows time-delay in information diffusion under the framework of independent cascade (IC) model, and has two kinds of parameters: the time-delay parameter and the diffusion parameter. We formulated the likelihood of obtaining the observed sequence data, and proposed an EM-like iterative method to obtain the parameter values by maximizing this likelihood. We first confirmed by using a complete graph that the proposed method outperforms a slightly modified existing method eGGLT in estimating correct parameters. Next, we showed by using three real world networks that the proposed method can much more accurately predict the influence degrees of the high-ranked nodes for the true ICTD model than the eGGLT method. Moreover, we demonstrated that it outperforms the eGGLT method and the conventional methods in social network analysis for ranking the influential nodes.

In conclusion, we blazed the path to learn a probabilistic information diffusion model over a network in a principled way. The IC model we used is the most basic, and there are other diffusion models [8]. A similar approach can be extended to these models, e.g., *linear threshold model* with time-delay, which will also be our future work.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

[1] N. Agarwal and H. Liu, Blogosphere: Research issues, tools, and application, *SIGKDD Explorations* **10** (2008), 18–31.

[2] R. Albert, H. Jeong, and A. L. Barabási, Error and attack tolerance of complex networks, *Nature* **406** (2000), 378–382.

[3] S. Brin and L. Page, The anatomy of a large-scale hypertextual web search engine, *Computer Networks and ISDN Systems* **30** (1998), 107–117.

[4] P. Domingos, Mining social networks for viral marketing, *IEEE Intelligent Systems* **20** (2005), 80–82.

[5] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, Critical phenomena in complex networks, *Reviews of Modern Physics* **80** (2008), 1275–1335.

[6] J. Goldenberg, B. Libai, and E. Muller, Talk of the network: A complex systems look at the underlying process of word-of-mouth, *Marketing Letters* **12** (2001), 211–223.

[7] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, Information diffusion through blogspace, in: *Proceedings of the 17th International World Wide Web Conference (WWW 2004)*, 2004, pp. 107–117.

[8] D. Kempe, J. Kleinberg, and E. Tardos, Maximizing the spread of influence through a social network, in: *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, 2003, pp. 137–146.

[9] M. Kimura, K. Saito, and R. Nakano, Extracting influential nodes for information diffusion on a social network, in: *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07)*, 2007, pp. 1371–1376.

[10] M. Kimura, K. Saito, and H. Motoda, Blocking links to minimize contamination spread in a social network, *ACM Transactions on Knowledge Discovery from Data* **3** (2009), Article 9.

[11] M. Kimura, K. Saito, R. Nakano, and H. Motoda, Finding influential nodes in a social network from information diffusion data, in: *Proceedings of the 2nd International Workshop on Social Computing, Behavioral Modeling and Prediction (SBP09)*, 2009, pp. 139–145.

[12] B. Klimt and Y. Yang, The Enron corpus: A new dataset for email classification research, in: *Proceedings of the 15th European Conference on Machine Learning (ECML 2004)*, 2004, pp. 217–226.

[13] J. Leskovec, L. Adamic, and B. A. Huberman, The dynamics of viral marketing, *ACM Transactions on the Web* **1** (2007), Article 5.

[14] M. E. J. Newman, S. Forrest, and J. Balthrop, Email networks and the spread of computer viruses, *Physical Review E* **66** (2002), Article 035101.

[15] M. E. J. Newman, The structure and function of complex networks, *SIAM Review* **45** (2003), 167–256.

[16] M. E. J. Newman and J. Park, Why social networks are different from other types of networks, *Physical Review E* **68** (2003), Article 036122.

[17] A. Y. Ng, A. X. Zheng, and M. I. Jordan, Link analysis, eigenvectors and stability, in: *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, 2001, pp. 903–910.

[18] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, *Nature* **435** (2005), 814–818.

[19] K. Saito, M. Kimura, and H. Motoda, Effective visualization of information diffusion process over complex networks, in: *Proceedings of the 2008 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2008)*, 2008, pp. 326–341.

[20] S. Wasserman, and K. Faust, *Social network analysis*, Cambridge University Press, 1994.

[21] D. J. Watts and P. S. Dodds, Influence, networks, and public opinion formation, *Journal of Consumer Research* **34** (2007), 441–458.

Tables

Table 1: Learning results for the complete graph dataset. Correct values: $r_{u,v} = 0.667$, $\kappa_{u,v} = 0.1$, for $\forall(u, v) \in E$.

Proposed method

M_0	mean($\boldsymbol{r}$)	std($\boldsymbol{r}$)	mean($\boldsymbol{\kappa}$)	std($\boldsymbol{\kappa}$)
20	0.641	0.254	0.103	0.058
40	0.688	0.180	0.101	0.041
60	0.677	0.166	0.101	0.037

eGGLT method

M_0	mean($\boldsymbol{r}$)	std($\boldsymbol{r}$)	mean($\boldsymbol{\kappa}$)	std($\boldsymbol{\kappa}$)
20	0.994	0.028	0.054	0.026
40	0.993	0.027	0.054	0.018
60	0.993	0.025	0.054	0.016

Table 2: Learning results for the blog network dataset. Correct values: $r_{u,v} = 0.667$, $\kappa_{u,v} = 0.1$, for $\forall(u, v) \in E$.

Proposed method

M_0	mean($\boldsymbol{r}$)	std($\boldsymbol{r}$)	mean($\boldsymbol{\kappa}$)	std($\boldsymbol{\kappa}$)
20	0.686	0.120	0.100	0.027
40	0.679	0.092	0.100	0.019
60	0.674	0.075	0.100	0.016

eGGLT method

M_0	mean($\boldsymbol{r}$)	std($\boldsymbol{r}$)	mean($\boldsymbol{\kappa}$)	std($\boldsymbol{\kappa}$)
20	0.756	0.135	0.096	0.029
40	0.750	0.119	0.096	0.022
60	0.746	0.112	0.096	0.019

Table 3: Learning results for the Enron network dataset. Correct values: $r_{u,v} = 0.667$, $\kappa_{u,v} = 0.1$, for $\forall (u, v) \in E$.

Proposed method

M_0	mean($\boldsymbol{r}$)	std($\boldsymbol{r}$)	mean($\boldsymbol{\kappa}$)	std($\boldsymbol{\kappa}$)
20	0.657	0.101	0.102	0.020
40	0.657	0.083	0.102	0.016
60	0.657	0.073	0.102	0.013

eGGLT method

M_0	mean($\boldsymbol{r}$)	std($\boldsymbol{r}$)	mean($\boldsymbol{\kappa}$)	std($\boldsymbol{\kappa}$)
20	0.856	0.137	0.082	0.030
40	0.855	0.134	0.082	0.028
60	0.854	0.133	0.082	0.028

Table 4: Learning results for the co-authorship network dataset. Correct values: $r_{u,v} = 0.667$, $\kappa_{u,v} = 0.2$, for $\forall (u, v) \in E$.

Proposed method

M_0	mean($\boldsymbol{r}$)	std($\boldsymbol{r}$)	mean($\boldsymbol{\kappa}$)	std($\boldsymbol{\kappa}$)
20	0.682	0.100	0.200	0.044
40	0.675	0.070	0.200	0.031
60	0.672	0.057	0.200	0.025

eGGLT method

M_0	mean($\boldsymbol{r}$)	std($\boldsymbol{r}$)	mean($\boldsymbol{\kappa}$)	std($\boldsymbol{\kappa}$)
20	0.690	0.100	0.209	0.048
40	0.682	0.070	0.209	0.034
60	0.680	0.057	0.209	0.028

Table 5: Mean influence-prediction errors $\mathcal{E}(k)$ of the proposed and eGGLT methods and the means of the true influence degrees $\mathcal{H}(k)$ for the three network datasets ($M_0 = 60$).

dataset	k	$\mathcal{E}(k)$ of proposed method	$\mathcal{E}(k)$ of eGGLT method	$\mathcal{H}(k)$
blog network	20	1.2	65.0	635.5
	200	1.8	71.0	581.9
Enron network	20	0.2	99.4	1500.7
	200	3.2	183.1	1485.1
co-authorship network	20	1.5	129.4	573.7
	200	2.1	105.1	415.1

Figure Captions

Fig. 1: Performance comparison for predicting the influence degrees of nodes in the case of $M_0 = 60$ for the blog network dataset.

Fig. 2: Performance comparison for predicting the influence degrees of nodes in the case of $M_0 = 60$ for the Enron network dataset.

Fig. 3: Performance comparison for predicting the influence degrees of nodes in the case of $M_0 = 60$ for the co-authorship network dataset.

Fig. 4: Performance comparison for ranking the influential nodes in the case of $M_0 = 60$ for the blog network dataset.

Fig. 5: Performance comparison for ranking the influential nodes in the case of $M_0 = 60$ for the Enron network dataset.

Fig. 6: Performance comparison for ranking the influential nodes in the case of $M_0 = 60$ for the co-authorship network dataset.

Figures

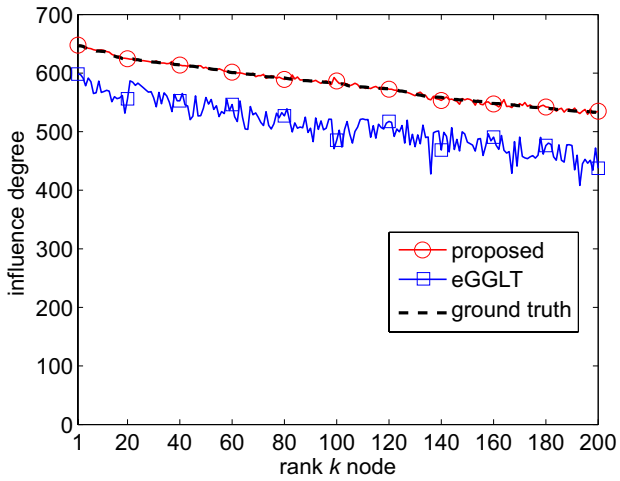


Figure 1:

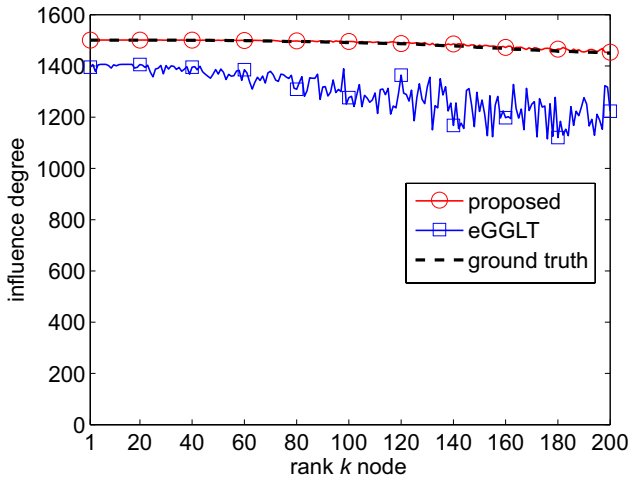


Figure 2:

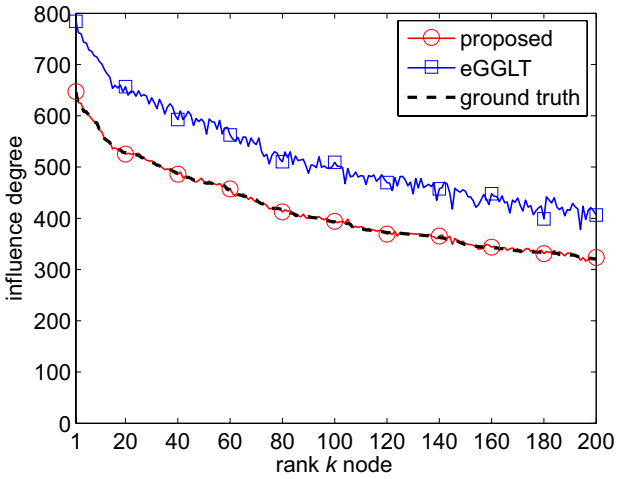


Figure 3:

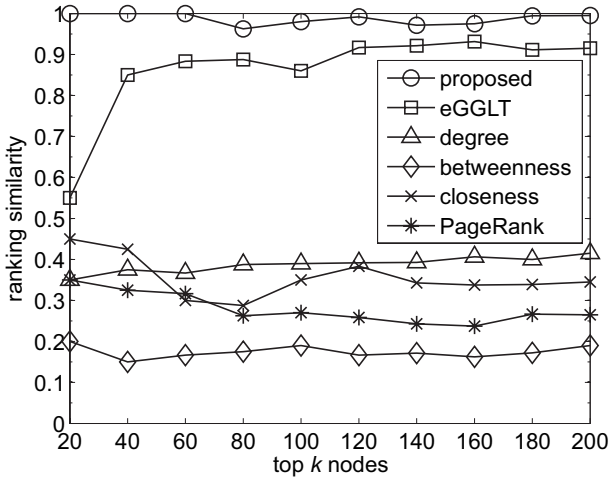


Figure 4:

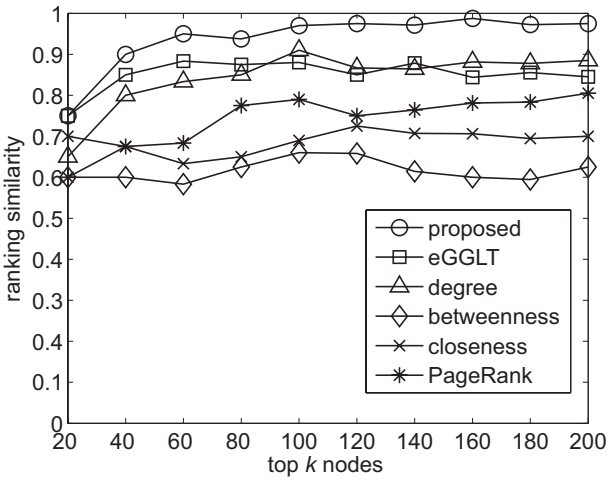


Figure 5:

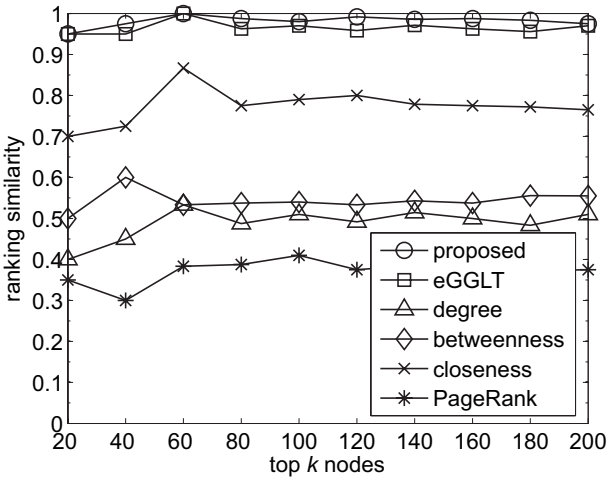


Figure 6:

Generative Models of Information Diffusion with Asynchronous Time-delay

Kazumi Saito K-SAITO@U-SHIZUOKA-KEN.AC.JP
School of Administration and Informatics
University of Shizuoka

Masahiro Kimura KIMURA@RINS.RYUKOKU.AC.JP
Department of Electronics and Informatics
Ryukoku University

Kouzou Ohara OHARA@IT.AOYAMA.AC.JP
Department of Integrated Information Technology
Aoyama Gakuin University

Hiroshi Motoda MOTODA@AR.SANKEN.OSAKA-U.AC.JP
Institute of Scientific and Industrial Research
Osaka University

Editor: Masashi Sugiyama and Qiang Yang

Abstract

We address the problem of formalizing an information diffusion process on a social network as a generative model in the machine learning framework so that we can learn model parameters from the observation. Time delay plays an important role in formulating the likelihood function as well as for the analyses of information diffusion. We identified that there are two different types of time delay: link delay and node delay. The former corresponds to the delay associated with information propagation, and the latter corresponds to the delay due to human action. We further identified that there are two distinctions of the way the activation from the multiple parents is updated: non-override and override. The former sticks to the initial activation and the latter can decide to update the time to activate multiple times. We formulated the likelihood function of the well known diffusion models: independent cascade and linear threshold, both enhanced with asynchronous time delay distinguishing the difference in two types of delay and two types of update scheme. Simulation using four real world networks reveals that there are differences in the spread of information diffusion and they strongly depend on the choice of the parameter values and the denseness of the network.

Keywords: Information diffusion, Social network, Maximum likelihood, Asynchronous time delay

1. Introduction

There have been tremendous interests in the phenomenon of influence that members of a social network can exert on other members and how the information propagates through the network. A variety of information that includes news, innovation, hot topics, ideas, opinions and even malicious rumors, propagates in the form of so-called "word-of-mouth" communications. Social networks (both real and virtual) are now recognized as an im-

portant medium for the spread of information and a considerable number of studies have been conducted (Newman et al., 2002; Newman, 2003; Gruhl et al., 2004; Domingos, 2005; Leskovec et al., 2006).

Basic models of information diffusion which are widely used in these studies are the *independent cascade (IC)* (Goldenberg et al., 2001; Kempe et al., 2003; Kimura et al., 2009) and the *linear threshold (LT)* (Watts, 2002; Watts and Dodds, 2007). They have been used to solve such problems as the *influence maximization problem* (Kempe et al., 2003; Kimura et al., 2010) and the *contamination minimization problem* (Kimura et al., 2009). Both models have parameters that need be specified in advance: diffusion probabilities for the IC model, and weights for the LT model. However, their true values are not known in practice. This poses yet another problem of estimating them from a set of information diffusion results that are observed as time-sequences of influenced (activated) nodes (Saito et al., 2009, 2010).

This problem fits in a well defined parameter estimation problem in machine learning setting, provided that a proper model is known. Thus, having a good generative model is crucial for this approach to be successful. One important factor that needs a special care is how to treat time delay in information diffusion. Diffusion process involves time evolution. The basic models deal with time by allowing nodes to change their states in a synchronous way at each discrete time step. No time delay is considered, or one can say that every action is uniformly delayed exactly by one discrete time step. However, it is indispensable to be able to cope with asynchronous time delay to do realistic analyses of information diffusion because, in the real world, information propagates along the continuous time axis, and time-delays can occur while information propagates by various reasons. Incorporating time-delay makes the time-sequence observation data structural, which makes the analyses of diffusion process difficult because it is not self-evident from the observed sequence data which node has activated which other node. What is observed is just a sequence of time when each node has been activated. Saito et al. (2009, 2010) have extended the basic IC and LT models to incorporate asynchronous time delay and successfully solved this parameter estimation problem by maximizing the likelihood function using a variant of EM algorithm, but they have not carefully examined that there are different types of time delay and node activation scheme[1].

In this paper, we revisit the generative model and carefully analyze what kind of time delay and activation scheme is considered realistic because, in general, the way the parameters are estimated depends on how the generative model is given. We identified that there are two different types of time delay: link delay and node delay. The former corresponds to the delay associated with information propagation, and the latter corresponds to the delay due to human action. We further identified that there are two types of the way the activation from the multiple parents is updated: non-override and override. The former sticks to the initial activation and the latter can decide to update the time to activate multiple times. We rigorously formulated the likelihood function of the IC and the LT models, extending them to incorporate asynchronous time delay with the difference in two types of time delay and two types of update scheme taken into account[2]. There are a total

1. Two examples explaining the different types of time delay are given in subsection 3.1.
2. We refer to asynchronous time delay versions of the IC and the LT models as the AsIC and AsLT models, respectively.

of three different models for each of the AsIC and the AsLT models, but the theoretical analysis revealed that particular combinations of time delay and update scheme result in the same likelihood function (with a minor notational difference) and it suffices to consider two different models for each. We performed how the difference in the time delay and the update scheme affects the information diffusion results as a function of time, varying the values of diffusion parameters using four real world networks. The simulation results reveal that there are differences in the spread of information diffusion and they strongly depend on the choice of the parameter values and the denseness of the network, confirming that it is important to distinguish the different types of time delay and update scheme. The results are well interpretable.

The paper is organized as follows. We revisit the basic information diffusion models in section 2 describing the likelihood functions. In section 3 we first explain different time delay types and update schemes, and then, based on these differences, derive the rigorous likelihood function for each of the possible combinations of these types and schemes for both the AsIC and AsLT models. We show the experimental result in subsection 5.2, and summarize the main conclusions in section 6.

2. Basic Information Diffusion Models

We mathematically model the spread of information through a directed network $G = (V, E)$ without self-links, where V and E ($\subset V \times V$) stand for the sets of all the nodes and links, respectively. For each node v in the network G, we denote $F(v)$ as a set of child nodes of v, *i.e.*, $F(v) = \{w; (v, w) \in E\}$. Similarly, we denote $B(v)$ as a set of parent nodes of v, *i.e.*, $B(v) = \{u; (u, v) \in E\}$. We call nodes *active* if they have been influenced with the information. In the following models, we assume that nodes can switch their states only from inactive to active, but not the other way around, and that, given an initial active node set S, only the nodes in S are active at an initial time.

2.1 Independent Cascade Model

We first recall the definition of the IC model according to Kempe et al. (2003). In the IC model, we specify a real value $\kappa_{u,v}$ with $0 < \kappa_{u,v} < 1$ for each link (u, v) in advance. Here $\kappa_{u,v}$ is referred to as the *diffusion probability* through link (u, v). The diffusion process unfolds in discrete time-steps $t \geq 0$, and proceeds from a given initial active set S in the following way. When a node u becomes active at time-step t, it is given a single chance to activate each currently inactive child node v, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time-step $t+1$. If multiple parent nodes of v become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

2.2 Linear Threshold Model

Next, we present the definition of the LT model. In this model, for every node $v \in V$, we specify a *weight* ($\omega_{u,v} > 0$) from its parent node u in advance such that $\sum_{u \in B(v)} \omega_{u,v} \leq 1$.

The diffusion process from a given initial active set S proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0, 1]$. At time-step t, an inactive node v is influenced by each of its active parent nodes, u, according to weight $\omega_{u,v}$. If the total weight from active parent nodes of v is no less than θ_v, that is, $\sum_{u \in B_t(v)} \omega_{u,v} \geq \theta_v$, then v will become active at time-step $t+1$. Here, $B_t(v)$ stands for the set of all the parent nodes of v that are active at time-step t. The process terminates if no more activations are possible.

2.3 Likelihood

As emphasized in section 1, our main focus is to formalize the information diffusion process as a generative model in machine learning problem setting. The generative model is a model of the world which can predict the future from the past, and the model must be consistent with the observation as much as possible. Thus, it is crucial to formulate the likelihood function as realistically as possible so that the model with these parameters (including $\kappa_{u,v}$ and $\omega_{u,v}$ above) that maximize the likelihood can best reflect the reality and generate the data close enough to the observation.

We denote an observed data set of M independent information diffusion results as $\{D_m;\ m = 1, \cdots, M\}$. Here, each D_m is a set of pairs of active nodes and their activation times in the mth diffusion result, $D_m = \{(u, t_{m,u}),\ (v, t_{m,v}),\ \cdots\}$. For each D_m, we denote the observed initial time by $t_m = \min\{t_{m,v};\ (v, t_{m,v}) \in D_m\}$, and the observed final time by $T_m \geq \max\{t_{m,v};\ (v, t_{m,v}) \in D_m\}$. Note that these are just sequences of $(u, t_{m,u})$ pairs and do not tell which parent node of u actually activated u. Further note that T_m is not necessarily equal to the final activation time. Hereafter, we express our observation data by $\mathcal{D}_M = \{(D_m, T_m);\ m = 1, \cdots, M\}$. For any $t \in [t_m, T_m]$, we set $C_m(t) = \{v;\ (v, t_{m,v}) \in D_m,\ t_{m,v} < t\}$. Namely, $C_m(t)$ is the set of active nodes before time t in the mth diffusion result. For convenience sake, we use C_m as referring to the set of all the active nodes in the mth diffusion result. Moreover, we define a set of non-active nodes with at least one active parent node for each by $\partial C_m = \{v;\ (u, v) \in E,\ u \in C_m,\ v \notin C_m\}$.

Next we formulate the likelihood function $\mathcal{L}(\mathcal{D}_M; \mathbf{\Theta})$, where $\mathbf{\Theta}$ denotes the parameters that we want to optimize by maximizing $\mathcal{L}$. Nodes in C_m are a part of the nodes in the graph G and those not in C_m have not been activated. Since non-activated nodes, unless they get activated, never activate the other non-activated node, we only need to consider nodes in C_m and ∂C_m. Thus, the likelihood function is basically described by the product of two factors, one representing the probabilities that nodes in C_m are activated at their respective times and the other representing the probabilities that nodes in ∂C_m have not been activated during the observed time period $[t_m, T_m]$.

The likelihood functions for the IC and the LT models take slightly different forms. $\mathcal{L}$ for the IC model is given by Equation (1), and $\mathcal{L}$ for the LT model is given by Equation (2).

$$\mathcal{L}(\mathcal{D}_M; \mathbf{\Theta}) = \prod_{m=1}^{M} \prod_{v \in C_m} (h_{m,v}\ g_{m,v}), \tag{1}$$

where $h_{m,v}$ is the probability density that the node v such that $v \in D_m$ with $t_{m,v} > 0$ for the mth diffusion result is activated at time $t_{m,v}$, and $g_{m,v}$ is the probability that a node v

fails to activate its child nodes for the mth diffusion result.

$$\mathcal{L}(\mathcal{D}_M;\boldsymbol{\Theta}) = \prod_{m=1}^{M}\left(\prod_{v\in C_m} h_{m,v}\right)\left(\prod_{v\in \partial C_m} g_{m,v}\right), \tag{2}$$

where the definition of $h_{m,v}$ is the same as above and $g_{m,v}$ is the probability that the node v is not activated within the observed time period $[t_m, T_m]$. The specific formulae of $h_{m,v}$ and $g_{m,v}$ for the IC model are

$$h_{m,v} = 1 - \prod_{u\in B(v)\cap \tilde{C}(t_{m,v})} (1-\kappa_{u,v}), \quad g_{m,v} = \prod_{w\in F(v)\setminus C(t_{m,v}+1)} (1-\kappa_{v,w}), \tag{3}$$

and those for the LT model are

$$h_{m,v} = \sum_{u\in B(v)\cap \tilde{C}(t_{m,v})} \omega_{u,v}, \quad g_{m,v} = 1 - \sum_{u\in B(v)\cap C_m} \omega_{u,v}, \tag{4}$$

where $\tilde{C}(t_{m,v}) = C(t_{m,v}) \setminus C(t_{m,v}-1)$. Note that Equations (3) have been described in **?**, and Equations (4) are special forms of the corresponding equations described in Saito et al. (2010).

3. Asynchronous Information Diffusion Models

In this section, we first explain notions of time-delay identifying two different types of time delay and two different types of the way the activation from the multiple parents is updated. Then, we derive the rigorous likelihood function for each of the possible combinations of these time-delay types and update schemes for asynchronous time delay versions of the IC and the LT models.

3.1 Notions of Time-delay

The basic information diffusion models briefly described in section 2 do not account for time delay. In reality it takes time for the information to diffuse by various reasons, and further, the way the delay takes place is asynchronous. Each parent u of a node v can be activated independently of the other parents in an asynchronous way and because the associated time delay from a parent to its child is different for every single pair, which parent u actually affects the node v in which order is more or less opportunistic. In case of the IC model which is sender-oriented, it may look more natural to attach the delay to the link, *i.e.*, when a node u is activated and is ready to send the information, it does not necessarily reach its child node v instantaneously but with some delay attached to the link (u, v). On the other hand, in case of the LT model which is receiver-oriented, it may look more natural to attach the delay to the node (receiver), *i.e.* when the sum of the weights from the active parents of a node v exceeds the threshold θ and the node v is ready to receive the information, it does not necessarily reach the node v instantaneously but with some delay attached to the node v. However, in both models information diffuses from a parent to its child and there is no reason to exclude other combinations than the above.

To explicate the information diffusion process in a more realistic setting, we think of two examples, one associated with blog posting and the other associated with electronic mailing. In case of blog posting, assume that some blogger u posts an article. Then it is natural to think that it takes some time before another blogger v comes to notice the posting. It is also natural to think that if the blogger v reads the article, he or she takes an action to respond (activated) because the act of reading the article is an active behavior. In this case, we can think that there is a delay in information diffusion from u to v but there is no delay in v taking an action. In case of electronic mailing, assume that someone u sends a mail to someone else v. It is natural to think that the mail is delivered to the receiver v instantaneously. However, this does not necessarily mean that v reads the mail as soon as it has been received because the act of receiving a mail is a passive behavior. In this case, we can think that there is no delay in information diffusion from u to v but there is a delay in v taking an action. Further, when v notices the mail, v may think to respond to it later. But before v responds, a new mail may arrive which needs a prompt response and v sends a mail immediately. We can think of this as an update of acting time[3]. These are just two examples, but it appears worth distinguishing the difference of these two kinds of time delay and update scheme (override of decision) in a more general setting.

In what follows we formulate the likelihood function distinguishing the difference of assumed time delay and override policy, and show that these distinctions indeed affect the form of the likelihood function. According to the discussion above, we define two types of delay: link delay and node delay. It is easiest to think that link delay corresponds to propagation delay and node delay corresponds to action delay. We further assume that they are mutually exclusive. This is a strong restriction as well as a strong simplification by necessity because the activation time we can observe is a sum of the two delays and we cannot distinguish between these two. Thus we have to choose either one of the two as occurring exclusively for the likelihood maximization to be feasible. In addition, we assume that there are two types of activation associated with time delay: non-override and override. The former sticks to the initial decision when to activate and the latter can decide to update (override) the time of activation multiple times each time one of the parents gets newly activated. Due to the mutual exclusiveness of link delay and node delay, override is only associated with node delay. As mentioned in section 1, we call the time delay versions of the IC and the LT models as Asynchronous Independent Cascade Model (AsIC) and Asynchronous Linear Threshold Model (AsLT), respectively.

In summary, node delay can go with either override or non-override, and link delay can only go with non-override. In the following subsections, we will derive $h_{m,v}$ and $g_{m,v}$ for each of the AsIC model and the AsLT model. We choose a delay-time δ from the exponential distribution with parameter r for the sake of convenience, but of course other distributions such as power-law and Weibull can be employed. The *time delay parameter* r is expressed explicitly as $r_{u,v}$ if it is link delay and r_v (or r_u) if it is node delay. Once the likelihood function is formalized, any optimization method can be used to find the best estimates of the parameter values. In practice, variants of EM algorithm has been shown to work satisfactorily (Saito et al., 2009, 2010).

3. Note that there are two actions here, reading and sending, but the activation time in the observed sequence data corresponds to the time v sends a mail

3.2 Asynchronous Independent Cascade Models

Link delay with non-override The diffusion process unfolds in continuous-time t, and proceeds from a given initial active set S in the following way. Suppose that a node u becomes active at time t. Then, u is given a single chance to activate each currently inactive child node v. If v has not been activated before time $t+\delta$, then u attempts to activate v, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time $t+\delta$. Under the continuous time framework, it is unlikely that v is activated simultaneously by its multiple parent nodes exactly at time $t+\delta$. So we ignore this possibility. The process terminates if no more activations are possible. Note that this delay is due to propagation delay. Once the node v receives the information, it instantaneously gets activated and there is no action delay in v.

We order the active parent node $u \in B(v) \cap C_m(t_{m,v})$ of a node v according to the time t_u it was activated: $B(v) \cap C_m(t_{m,v}) = \{u_1, u_2, ..., u_K\}$ such that $t_{u_1} < t_{u_2} < ... < t_{u_K}$.

The probability density $h_{m,v}$ is the sum of the probability density that u_i activates v but all the other $u_j, j \neq i$ fail to activate v over all i $(i = 1, 2, ..., K)$.

$$\begin{aligned} h_{m,v} &= \sum_{k=1}^{K} \kappa_{u_k,v} r_{u_k,v} \exp(-r_{u_k,v}(t_{m,v} - t_{m,u_k})) \\ &\quad \times \prod_{i=1, i\neq k}^{K} (1 - \int_{t_{m,u_i}}^{t_{m,v}} \kappa_{u_i,v} r_{u_i,v} \exp(-r_{u_i,v}(t - t_{m,u_i}))dt) \\ &= \sum_{k=1}^{K} \kappa_{u_k,v} r_{u_k,v} \exp(-r_{u_k,v}(t_{m,v} - t_{m,u_k})) \\ &\quad \times \prod_{i=1, i\neq k}^{K} (\kappa_{u_i,v} \exp(-r_{u_i,v}(t_{m,v} - t_{m,u_i})) + (1 - \kappa_{u_i,v})) \,. \end{aligned} \tag{5}$$

The probability $g_{m,v}$ is given by

$$\begin{aligned} g_{m,v} &= \prod_{w \in F(v) \setminus C_m} \left(1 - \int_{t_{m,v}}^{T_m} \kappa_{v,w} \exp(-r_{v,w}(t - t_{m,v}))dt\right) \\ &= \prod_{w \in F(v) \setminus C_m} (\kappa_{v,w} \exp(-r_{v,w}(T_m - t_{m,v})) + (1 - \kappa_{v,w})) \,. \end{aligned} \tag{6}$$

Note that the formulation in Saito et al. (2009) corresponds to this category.

Node delay with non-override The difference of diffusion process from *Link delay with non-override* is that there is no delay in propagating the information to the node v from the node u, but there is a delay δ before the node v gets actually activated. Assume that it is the node u_i that first succeeded in activating the node v (more precisely satisfying the activation condition). Since there is no link delay, it must be the case that all the other parents that had become active before t_{u_i} must have failed in activating v (more precisely satisfying the activation condition). Since the node v decides when to actually activate itself at the time the node u_i succeeded in satisfying the activation condition and would not

change its mind, other nodes which may possibly activate the node v at a later time can do nothing on the node v. Thus, the probability density $h_{m,v}$ is given by

$$h_{m,v} = \sum_{k=1}^{K} \kappa_{u_k,v} \prod_{i=1}^{k-1} (1-\kappa_{u_i,v}) r_v \exp(-r_v(t_{m,v}-t_{m,u_k})). \quad (7)$$

The probability $g_{m,v}$ is the same as Equation (6).

Node delay with override The difference of diffusion process from *Node delay with non-override* is that here the actual activation time is allowed to be updated. For example, suppose that the node u_i first succeeded in satisfying the activation condition of the node v and the node v decided to activate itself at time $t_{u_i}+\delta_i$. At some time later but before $t_{u_i}+\delta_i$, other parent u_j also succeeded in satisfying the activation condition of the node v. Then the node v is allowed to change its actual activation time to time $t_{u_j}+\delta_j$ which may be before $t_{u_i}+\delta_i$. Thus, the probability density $h_{m,v}$ is given by

$$\begin{aligned} h_{m,v} &= \sum_{k=1}^{K} \kappa_{u_k,v} r_v \exp(-r_v(t_{m,v}-t_{m,u_k})) \\ &\quad \times \prod_{i=1,i\neq k}^{K} (1 - \int_{t_{m,u_i}}^{t_{m,v}} \kappa_{u_i,v} r_v \exp(-r_v(t-t_{m,u_i}))dt) \\ &= \sum_{k=1}^{K} \kappa_{u_k,v} r_v \exp(-r_v(t_{m,v}-t_{m,u_k})) \\ &\quad \times \prod_{i=1,i\neq k}^{K} (\kappa_{u_i,v} \exp(-r_v(t_{m,v}-t_{m,u_i})) + (1-\kappa_{u_i,v})) . \end{aligned} \quad (8)$$

The probability $g_{m,v}$ is the same as Equation (6).

3.3 Asynchronous Linear Threshold Models

Link delay with non-override The diffusion process unfolds in continuous-time t, and proceeds from a given initial active set S in the following way. When a node u_i is activated at t_{u_i}, it exerts its effect on its child node v with a delay δ_i. Suppose that the accumulated weight from the active parents of node v has become no less than θ_v at time t for the first time. The node v becomes active without any delay (no node delay) and exerts its effect on its child with a delay δ_v. Because there is no override, there is no update of the activation time of the node v. This process is repeated until no more activations are possible.

It is to be noted that because δ_i is a random variable, $t_{u_i}+\delta_i$ is not monotonic with respect to i even though u_i is ordered according to the activation time t_{u_i}. We define a new ordering of the parent node i according to the time $t_{u_i}+\delta_i$ that it exerts its effect on its child v. Suppose the node v first become activated for i of this new ordering. Then the threshold θ_v is between $\sum_{j=1}^{i-1}\omega_{u_j,v}$ and $\sum_{j=1}^{i-1}\omega_{u_j,v}+\omega_{u_i,v}$. Since θ_v is uniformly distributed, the probability that θ_v is chosen from this range is $\omega_{u_i,v}$. Thus, the probability density $h_{m,v}$

that the node v is activated at time $t_{m,v}$ can be expressed as

$$h_{m,v} = \sum_{k=1}^{K} \omega_{u_k,v} r_{u_k,v} \exp(-r_{u_k,v}(t_{m,v} - t_{m,u_k})). \tag{9}$$

The probability $g_{m,v}$ that a node v is not activated with the observed time period $[t_m, T_m]$ is given by

$$\begin{aligned} g_{m,v} &= 1 - \sum_{k=1}^{K} \omega_{u_k,v} \int_{t_{m,u_k}}^{T_m} r_{u_k,v} \exp(-r_{u_k,v}(t - t_{m,u_k}))dt \\ &= 1 - \sum_{k=1}^{K} \omega_{u_k,v}(1 - \exp(-r_{u_k,v}(T_m - t_{m,u_k}))). \end{aligned} \tag{10}$$

Node delay with non-override The difference of diffusion process from *Link delay with non-override* is that as soon as the parent node u_i is activated, its effect is immediately exerted to its child v. The delay depends on the node v's choice.

Suppose the node v first became activated for i of the parent ordering according to the time t_{u_i}. Then by the same reasoning as before, the threshold θ_v is between $\sum_{j=1}^{i-1} \omega_{u_j,v}$ and $\sum_{j=1}^{i-1} \omega_{u_j,v} + \omega_{u_i,v}$, and the probability density $h_{m,v}$ can be expressed as

$$h_{m,v} = \sum_{k=1}^{K} \omega_{u_k,v} r_v \exp(-r_v(t_{m,v} - t_{m,u_k})). \tag{11}$$

The probability $g_{m,v}$ is the same as Equation (10). Note that the formulation in Saito et al. (2010) corresponds to this category.

Node delay with override The difference of diffusion process from *Node delay with non-override* is that multiple updates of the activation time of the node v is allowed. Suppose that the node v first became activated by receiving the effect of the parent u_k. All the parents that have become activated after that can still influence the updates. Considering the probability that node u_i's effect eventually leads the node v's activation at a time later than $t_{m,v}$, the probability density that the node v is activated at time $t_{m,v}$ by one of its parent nodes which get activated later than u_k for which the threshold is first exceeded is

$$\begin{aligned} h_{m,u_k,v} &= \omega_{u_k,v} \sum_{j=k}^{K} r_v \exp(-r_v(t_{m,v} - t_{m,u_j})) \prod_{i=k, i\neq j}^{K} \int_{t_{m,v}}^{\infty} r_v \exp(-r_v(t - t_{m,u_i}))dt \\ &= \omega_{u_k,v}(K - k + 1) r_v \prod_{i=k}^{K} \exp(-r_v(t_{m,v} - t_{m,u_i})). \end{aligned} \tag{12}$$

Thus, finally we obtain

$$h_{m,v} = \sum_{k=1}^{K} h_{m,u_k,v}. \tag{13}$$

The probability $g_{m,v}$ is the same as Equation (10).

4. Properties of Asynchronous Time-delay models

In this section, we describe some properties of asynchronous time-delay models in terms of the expected influence degree and behavioral analysis.

4.1 Expected Influence Degree

The expected influence degree of each node v, which is defined by the expected length of information diffusion sequence starting from the node v, plays a crucial role to solve the several important problems such as influence maximization and contamination minimization. Here we can easily see that for the same diffusion parameters $\kappa_{u,v}$ (or $\omega_{u,v}$), the expected influence degree of each node obtained by the basic IC (or LT) model is equal to the one obtained by any variant of AsIC (or AsLT) model after a substantially large time has passed. This is because what the asynchronous time models are doing is simply controlling the activation time of each node in relative to the basic models, but the asymptotic values of the expected influence degree remain the same. Thus, for the purpose of obtaining the expected influence degree, it suffices to use the basic models and we can apply any kind of efficient methods such as the bond percolation method (Kimura et al., 2010).

However, for some applications, such as the maximization of information spread to promote sales during a certain period of time, estimating the expected influence degree at a specific time or at a specific time interval may become very important and essential, *i.e.* a transient phenomenon becomes important. In particular, we can naturally conceive that each variant of the asynchronous time-delay models shows a different effect of time-delay on information diffusion. Since it is quite difficult to obtain analytical results, we attempt to clarify such effect by performing the experimental evaluation shown in the next section.

4.2 Behavioral Analysis

It has been shown in Saito et al. (2009, 2010) that behavioral analysis can reveal intrinsic characteristics of a given information diffusion sequence, under the assumption that people behave quite similarly for the same topic of information diffusion. Thus far, we have assumed that $\mathbf{\Theta}$ can vary with respect to nodes and links but is independent of the topic of information diffused. However, as predicted, they may be sensitive to the topic. If we place a constraint that $\mathbf{\Theta}$ depends only on topics but not on nodes and links of the network G, we can assign a different m to a different topic. Under this setting, we can set $r_{m,u,v} = r_m$ or $r_{m,v} = r_m$, $\kappa_{m,u,v} = \kappa_m$ and $\omega_{m,u,v} = \omega_m = q_m|B(v)|^{-1}$ for any link $(u, v) \in E$ and for any node $v \in V$. Here note that the newly introduced parameter $q_m (0 < q_m < 1)$ is the one which corresponds to κ in the AsIC model and $\omega_{v,v} = 1 - q_m$. Using each pair of the estimated parameters, (r_m, κ_m) for the AsIC model and (r_m, q_m) for the AsLT model, we can discuss which model is more appropriate for each topic, and analyze the behavior of people with respect to the topics of information by simply plotting them as a point in the two-dimensional space. The validity of the above assumption has been confirmed using a real diffusion dataset in blogsphare as exemplified in Saito et al. (2009, 2010).

Looking through the results in the previous subsections, we note that in case of the AsIC model $h_{m,v}$ takes the same form for *Link delay with non-override* and *Node delay with override*, and in case of the AsLT model $h_{m,v}$ takes the same form for *Node delay with*

non-override and *Link delay with non-override*. This means that in terms of the behavioral analysis as is explained above, interestingly these respective two different time delay models give the same results.

5. Evaluation of Effect of Time-delay on Information Diffusion

As mentioned earlier, we empirically evaluate the effect of the difference in the time-delay models on information diffusion using four real world networks. To this end, we introduce the following unified measure to quantify the average speed of propagation for networks of different sizes, as well as with various parameter settings in the information diffusion models.

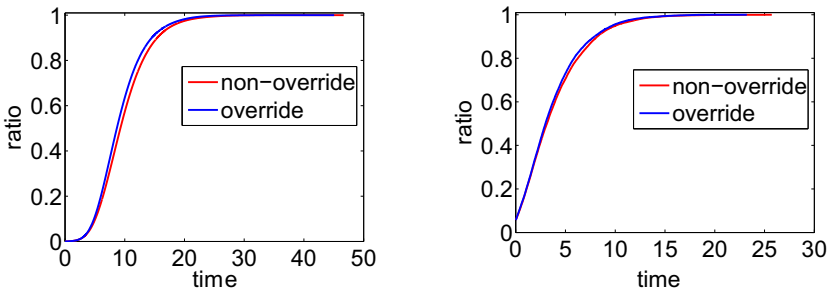
$$p(t) \quad = \quad \frac{\sum_{m=1}^{M} |\{(v_m, t_{m,v}) \in \mathcal{D}_m; t_{m,v} \leq t\}|}{\sum_{m=1}^{M} |\mathcal{D}_m|}. \tag{14}$$

For a given set of information results $\{D_m;\ m = 1, \cdots, M\}$ and a specified time t, this measure gives the expected ratio of the number of activated nodes until t to that of the total activated nodes. In our experiments, the initial and final times were set to $t_m = 0$ and $T_m = \infty$, respectively, for each information diffusion sequence m.

5.1 Network Data

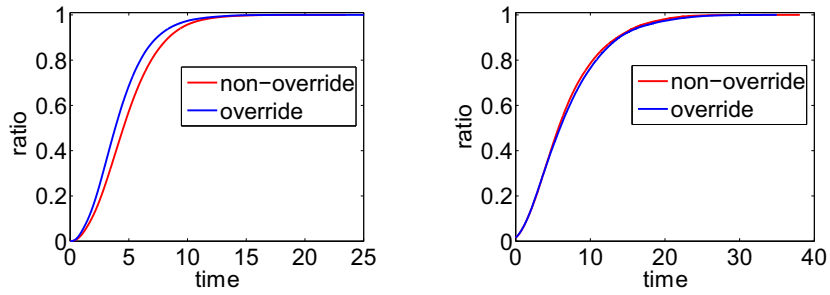
We employed four datasets of large real networks (all bidirectionally connected) and used their structures to generate diffusion data. The first one is a coauthorship network used in Palla et al. (2005) and has 12,357 nodes and 38,896 directed links (the coauthorship network). The second one is a trackback network of Japanese blogs used in Kimura et al. (2009) and has 12,047 nodes and 79,920 directed links (the blog network). The third one is a network derived from the Enron email dataset (Klimt and Yang, 2004) by extracting the senders and the recipients and linking those that had bidirectional communications. It has 4,254 nodes and 44,314 directed links (the Enron network). The fourth one is a network of people derived from the "list of people" within Japanese Wikipedia, also used in Kimura et al. (2009), and has 9,481 nodes and 245,044 directed links (the Wikipedia network).

As a practical situation, we evaluated the information diffusion models in the framework of behavioral analyses. Then, as explained in the previous section, link delay with non-override and node delay with override are indistinguishable for the AsIC model, while link delay and node delay both with non-override are indistinguishable for the AsLT model. Thus, we focused on node delay and evaluated the effect of override and non-override for both the AsIC and AsLT models, *i.e.*, $\kappa_{u,v} = \kappa$, $r_v = r$ for AsIC, and $\omega_{u,v} = q|B(v)|^{-1}$, $r_v = r$ for AsLT. In our preliminary experiments, changing the parameter r worked only for scaling the time axis of the diffusion results. Thus, we fixed its value at 1 ($r = 1$) for all cases and evaluated the effects of other diffusion parameters (κ for the AsIC model and q for the AsLT model). We prepared two different values (small and big) for both κ and q for each network. The values for κ were chosen to be the double and the half of the baseline value which is defined by $1/\bar{d}$, where $\bar{d}$ is the mean out-degree of a network. Each baseline value of κ becomes 0.2 for the coauthorship network, 0.1 for the blog and Enron networks, and 0.04 for the Wikipedia network. The values for q were set to 1 and 0.5, respectively



(a) $\kappa = 0.4$ (b) $\kappa = 0.1$

Figure 1: Results for the AsIC models in the coauthor network.



(a) $\kappa = 0.2$ (b) $\kappa = 0.05$

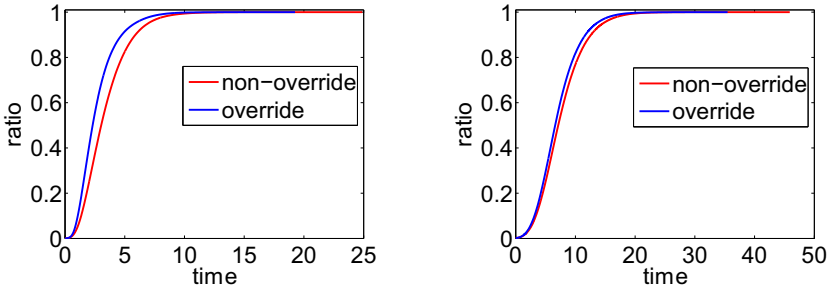
Figure 2: Results for the AsIC models in the blog network.

and used for all networks. Eventually, $M = 1,000$ information diffusion results with the sequence length of at least 10 were generated for each of these parameter values for each network, randomly selecting an initial active node for each diffusion result.

5.2 Experimental Results

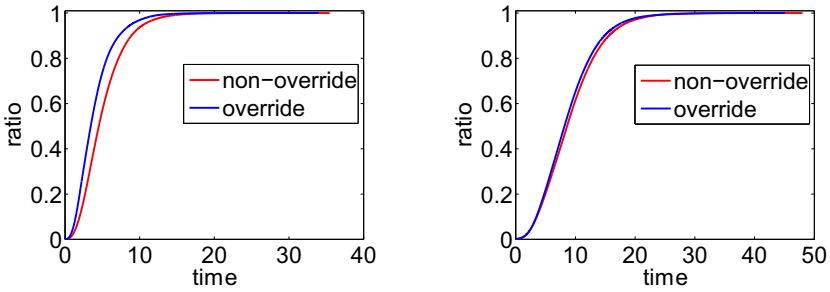
In Figures 1, 2, 3, and 4, we show experimental results for the AsIC models by using the respective networks: coauthorship, blog, Enron, and Wikipedia. We note that it takes longer for the ratio to converge to 1.0 in Figure 1a than in Figure 1b although the diffusion probability κ is larger in Figure 1a than in Figure 1b. This does not necessarily mean that the diffusion is slower for the case where the diffusion probability is larger. The main reason is due to the difference of the number of active nodes. A larger diffusion probability generates a longer diffusion sequence which, in turn, takes a longer time. This tendency is not clear for the other figures because the diffusion probability is at most $\kappa = 0.2$. The same is true for the AsLT model.

We further see that there is very little difference between non-override and override schemes when the diffusion parameter is small (half of the baseline value), but the difference becomes larger and the speed of information diffusion becomes faster for override scheme when the diffusion parameter is large (double of the baseline value). Here note that we chose each diffusion parameter according to the ratio of the numbers of nodes to links. This means that the value for κ is set to be reversely proportional to the network denseness and



(a) $\kappa = 0.2$ (b) $\kappa = 0.05$

Figure 3: Results for the AsIC models in the enron network.



(a) $\kappa = 0.08$ (b) $\kappa = 0.02$

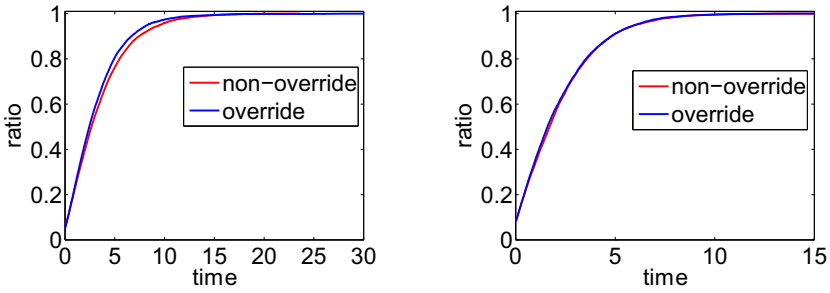
Figure 4: Results for the AsIC models in the wiki network.

the diffusion properties are supposed to be similar to each other. Thus, the results indicate that the effects of time-delay on the information diffusion models become much larger for a denser network when the diffusion parameter is large although no such difference is observed among the networks of different denseness when the diffusion parameter is small.

In Figures 5, 6, 7, and 8, we show experimental results for the AsLT models by using the respective networks: coauthorship, blog, Enron, and Wikipedia. Similarly to the AsIC model, here again we see hardly the difference between non-override and override schemes when the diffusion parameters are small ($q = 0.5$), but we do see that there is the difference between the two schemes and the speed of information diffusion becomes faster for override scheme when the diffusion parameters are large ($q = 1$). The effect of the difference of the network denseness is similar to the results for the AsIC model. In particular, we observe this difference is larger in the order of the Wikipedia, Enron, blog, and coauthorship networks. Here note that this order coincides with the descending order of their average degrees, *i.e.*, denseness of the network. This suggests that the effects of time-delay on the information diffusion models become much larger for a denser network when the diffusion parameter is large.

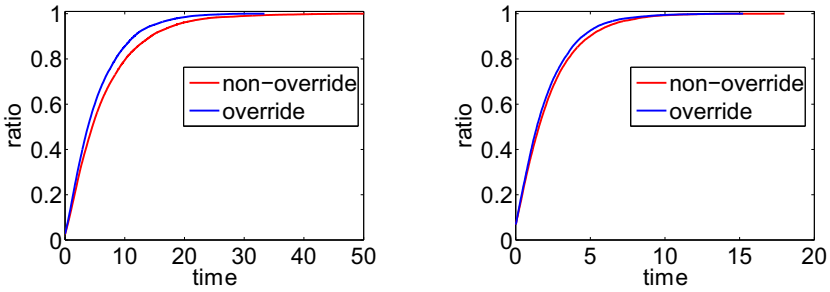
6. Conclusion

We formalized an information diffusion process as a generative model in the machine learning framework. In particular, we emphasized that the treatment of the time delay is important



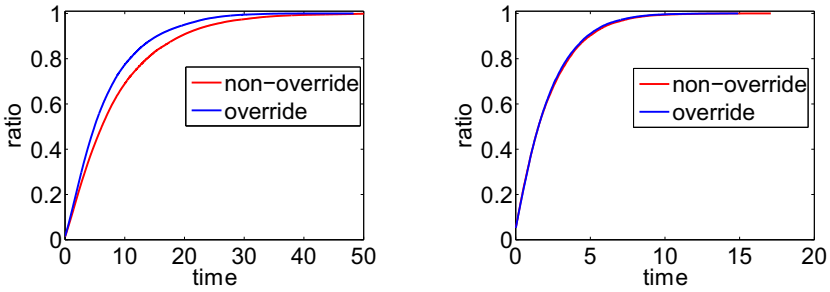
(a) $q = 1$ (b) $q = 0.5$

Figure 5: Results for the AsLT models in the coauthor network.



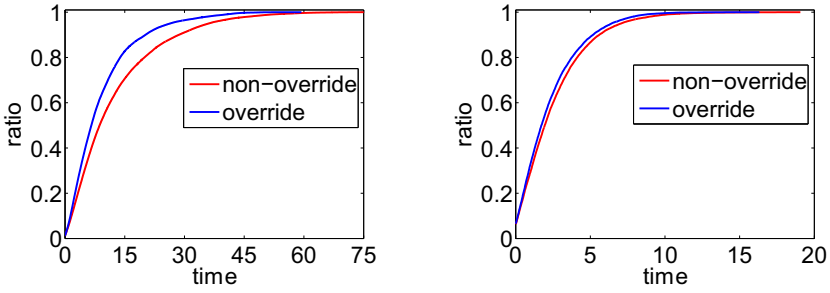
(a) $q = 1$ (b) $q = 0.5$

Figure 6: Results for the AsLT models in the blog network.



(a) $q = 1$ (b) $q = 0.5$

Figure 7: Results for the AsLT models in the enron network.



(a) $q = 1$ (b) $q = 0.5$

Figure 8: Results for the AsLT models in the wiki network.

in deriving the likelihood function. Diffusion comes with the notion of time and the probabilistic nature of the diffusion model hides the time delay structure from the surface of the observed sequence data, and makes the analysis difficult. We identified that there are two different types of time delay which we named link delay and node delay. The former corresponds to the delay associated with information propagation, and the latter corresponds to the delay associated with human action. We further identified that there are two different schemes of the way the activation from the multiple parents is updated which we named non-override and override. The former sticks to the initial activation and the latter can decide to update the time to activate multiple times. We applied these different notions of time delay to the well known basic information diffusion models: independent cascade (IC) and linear threshold (LT), and formalized asynchronous time delay versions of the IC and the LT models (AsIC and AsLT). We then derived a rigorous likelihood function for the feasible combinations of the time delay and update scheme for each of the AsIC and the AsLT models. There are a total of three different models for each diffusion models (AsIC and AsLT), but the theoretical analysis revealed that particular combinations of time delay and update scheme result in the same likelihood function (with a minor notational difference) and it is sufficient to consider two different models for each. We performed experiments to see how the difference in the time delay and the update scheme affects the information diffusion results as a function of time, varying the values of diffusion parameters using four real world networks. The simulation results reveal that there are differences in the spread of information diffusion and they strongly depend on the choice of the parameter values and the denseness of the network. We confirmed that it is important to distinguish the different types of time delay and update scheme in particular for a dense network that has a large information diffusion parameter value.

References

P. Domingos. Mining social networks for viral marketing. *IEEE Intelligent Systems*, 20: 80–82, 2005.

J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12:211–223, 2001.

D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. *SIGKDD Explorations*, 6:43–52, 2004.

D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, pages 137–146, 2003.

M. Kimura, K. Saito, and H. Motoda. Blocking links to minimize contamination spread in a social network. *ACM Transactions on Knowledge Discovery from Data*, 3:9:1–9:23, 2009.

M. Kimura, K. Saito, R. Nakano, and H. Motoda. Extracting influential nodes on a social network for information diffusion. *Data Mining and Knowledge Discovery, Springer*, 20: 70–97, 2010.

B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *Proceedings of the 2004 European Conference on Machine Learning (ECML'04)*, pages 217–226, 2004.

J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06)*, pages 228–237, 2006.

M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45: 167–256, 2003.

M. E. J. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 66:035101, 2002.

G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

K. Saito, M. Kimura, K. Ohara, and H. Motoda. Learning continuous-time information diffusion model for social behavioral data analysis. In *Proceedings of the 1st Asian Conference on Machine Learning (ACML2009)*, pages 322–337, 2009.

K. Saito, M. Kimura, K. Ohara, and H. Motoda. Behavioral analyses of information diffusion models by observed data of social network. In *Proceedings of the International Workshop on Social Computing and Behavioral Modeling (SBP10)*, pages 149–158, 2010.

D. J. Watts. A simple model of global cascades on random networks. *Proceedings of National Academy of Science, USA*, 99:5766–5771, 2002.

D. J. Watts and P. S. Dodds. Influence, networks, and public opinion formation. *Journal of Consumer Research*, 34:441–458, 2007.

Discovery of Super-Mediators of Information Diffusion in Social Networks

Kazumi Saito[1], Masahiro Kimura[2], Kouzou Ohara[3], and Hiroshi Motoda[4]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[3] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We address the problem of discovering a different kind of influential nodes, which we call "super-mediator", i.e. those nodes which play an important role to pass the information to other nodes, and propose a method for discovering super-mediators from information diffusion samples without using a network structure. We divide the diffusion sequences in two groups (lower and upper), each assuming some probability distribution, find the best split by maximizing the likelihood, and rank the nodes in the upper sequences by the F-measure. We apply this measure to the information diffusion samples generated by two real networks, identify and rank the super-mediator nodes. We show that the high ranked super-mediators are also the high ranked influential nodes when the diffusion probability is large, i.e. the influential nodes also play a role of super-mediator for the other source nodes, and interestingly enough that when the high ranked super-mediators are different from the top ranked influential nodes, which is the case when the diffusion probability is small, those super-mediators become the high ranked influential nodes when the diffusion probability becomes larger. This finding will be useful to predict the influential nodes for the unexperienced spread of new information, e.g. spread of new acute contagion.

1 Introduction

There have been tremendous interests in the phenomenon of influence that members of social network can exert on other members and how the information propagates through the network. Social networks (both real and virtual) are now recognized as an important medium for the spread of information. A variety of information that includes news, innovation, hot topics, ideas, opinions and even malicious rumors, propagates in the form of so-called "word-of-mouth" communications. Accordingly, a considerable amount of studies has been made for the last decade [1–20].

Among them, widely used information diffusion models are the *independent cascade (IC)* [1, 8, 13] and the *linear threshold (LT)* [4, 5] and their variants [14, 15, 6, 16–18]. These two models focus on different information diffusion aspects. The IC model is sender-centered and each active node *independently* influences its inactive neighbors with given diffusion probabilities. The LT model is receiver-centered and a node is influenced by its active neighbors if their total weight exceeds the threshold for the node. Which model is more appropriate depends on the situation and selecting the appropriate one is not easy [18].

The major interests in the above studies are finding influential nodes, i.e. finding nodes that play an important role of spreading information as much as possible. This problem is called *influence maximization problem* [8, 10]. The node influence can only be defined as the expected number of active nodes (nodes that have become influenced due to information diffusion) because the diffusion phenomenon is stochastic, and estimating the node influence efficiently is still an open problem. Under this situation, solving an optimal solution, i.e. finding a subset of nodes of size K that maximizes the expected influence degree with K as a parameter, faces with combinatorial explosion problem and, thus, much of the efforts has been directed to finding algorithms to efficiently estimate the expected influence and solve this optimization problem. For the latter, a natural solution is to use a greedy algorithm at the expense of optimality. Fortunately, the expected influence degree is submodular, i.e. its marginal gain diminishes as the size K becomes larger, and the greedy solution has a lower bound which is 63% of the true optimal solution [8]. Various techniques to further reduce the computational cost have been attempted including bond percolation [10] and pruning [14] for the former, and lazy evaluation [21], burnout [15] and heuristics [22] for the latter.

Expected influence degree is approximated by the empirical mean of the influence degree of many independent information diffusion simulations, and by default it has been assumed that the degree distribution is Gaussian. However, we noticed that this assumption is not necessarily true, which motivated to initiate this work. In this paper, we address the problem of discovering a different kind of influential nodes, which we call ”super-mediator”, i.e. those nodes which play an important role in passing the information to other nodes, and try to characterize such nodes, and propose a method for discovering super-mediator nodes from information diffusion sequences (samples) without using a network structure. We divide the diffusion samples in two groups (lower and upper), each assuming some probability distribution, find the best split by maximizing the likelihood, and rank the nodes in the upper sequences by the F-measure (more in subsection 3.2).

We tested our assumption of existence of super-mediators using two real networks[1] and investigated the utility of the F-measure. As before, we assume that information diffusion follows either the independent cascade (IC) model or the linear threshold (LT) model. We first analyze the distribution of influence degree averaged over all the initial nodes[2] based on the above diffusion models, and empirically show that it becomes a

[1] Note that we use these networks only to generate the diffusion sample data, and thus are not using the network structure for the analyses.

[2] Each node generates one distribution, which is approximated by running diffusion simulation many times and counting the number of active nodes at the end of simulation.

power-law like distribution for the LT model, but it becomes a mixture of two distributions (power-law like distribution and lognormal like distributions) for the IC model. Based on this observation, we evaluated our super-mediator discovery method by focusing on the IC model. It is reasonable to think that the super mediators themselves are the influential nodes, and we show empirically that the high ranked super-mediators are indeed the high ranked influential nodes, i.e. the influential nodes also play a role of super-mediator for the other source nodes, but this is true only when the diffusion probability is large. What we found more interesting is that when the high ranked super-mediators are different from the top ranked influential nodes, which is the case when the diffusion probability is small, those super-mediators become the high ranked influential nodes when the diffusion probability becomes larger. We think that this finding is useful to predict the influential nodes for the unexperienced spread of new information from the known experience, e.g. spread of new acute contagion from the spread of known moderate contagion for which there are abundant data.

The paper is organized as follows. We start with the brief explanation of the two information diffusion models (IC and LT) and the definition of influence degree in section 2, and then describe the discovery method based on the likelihood maximization and F-measure in section 3. Experimental results are detailed in section 4 together with some discussion. We end this paper by summarizing the conclusion in section 5.

2 Information Diffusion Models

We mathematically model the spread of information through a directed network $G = (V, E)$ without self-links, where V and E ($\subset V \times V$) stand for the sets of all the nodes and links, respectively. For each node v in the network G, we denote $F(v)$ as a set of child nodes of v, i.e. $F(v) = \{w; (v, w) \in E\}$. Similarly, we denote $B(v)$ as a set of parent nodes of v, i.e. $B(v) = \{u; (u, v) \in E\}$. We call nodes *active* if they have been influenced with the information. In the following models, we assume that nodes can switch their states only from inactive to active, but not the other way around, and that, given an initial active node set H, only the nodes in H are active at an initial time.

2.1 Independent Cascade Model

We recall the definition of the IC model according to [8]. In the IC model, we specify a real value $p_{u,v}$ with $0 < p_{u,v} < 1$ for each link (u, v) in advance. Here $p_{u,v}$ is referred to as the *diffusion probability* through link (u, v). The diffusion process unfolds in discrete time-steps $t \geq 0$, and proceeds from a given initial active set H in the following way. When a node u becomes active at time-step t, it is given a single chance to activate each currently inactive child node v, and succeeds with probability $p_{u,v}$. If u succeeds, then v will become active at time-step $t + 1$. If multiple parent nodes of v become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

2.2 Linear Threshold Model

In the LT model, for every node $v \in V$, we specify a *weight* ($\omega_{u,v} > 0$) from its parent node u in advance such that $\sum_{u \in B(v)} \omega_{u,v} \leq 1$. The diffusion process from a given initial active set H proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0, 1]$. At time-step t, an inactive node v is influenced by each of its active parent nodes, u, according to weight $\omega_{u,v}$. If the total weight from active parent nodes of v is no less than θ_v, that is, $\sum_{u \in B_t(v)} \omega_{u,v} \geq \theta_v$, then v will become active at time-step $t+1$. Here, $B_t(v)$ stands for the set of all the parent nodes of v that are active at time-step t. The process terminates if no more activations are possible.

2.3 Influence Degree

For both models on G, we consider information diffusion from an initially activated node v, i.e. $H = \{v\}$. Let $\varphi(v; G)$ denote the number of active nodes at the end of the random process for either the IC or the LT model on G. Note that $\varphi(v; G)$ is a random variable. We refer to $\varphi(v; G)$ as the *influence degree* of node v on G. Let $\mathcal{E}(v; G)$ denote the expected number of $\varphi(v; G)$. We call $\mathcal{E}(v; G)$ the *expected influence degree* of node v on G. In theory we can simply estimate $\mathcal{E}$ by the simulations based on either the IC or the LT model in the following way. First, a sufficiently large positive integer M is specified. Then, the diffusion process of either the IC or the LT model is simulated from the initially activated node v, and the number of active nodes at the end of the random process, $\varphi(v; G)$, is calculated. Last, $\mathcal{E}(v; G)$ for the model is estimated as the empirical mean of influence degrees $\varphi(v; G)$ that are obtained from M such simulations.

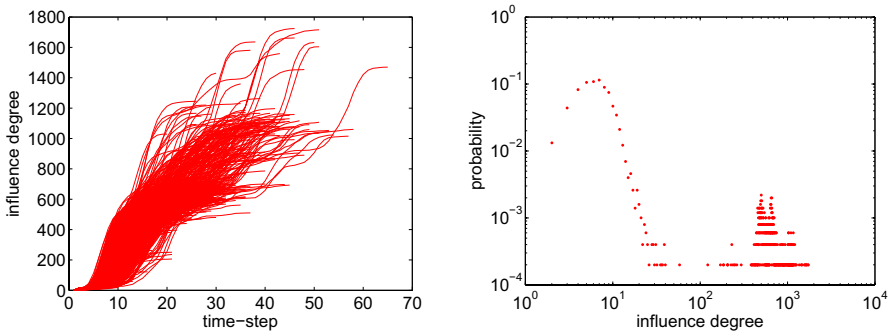
From now on, we use $\varphi(v)$ and $\mathcal{E}(v)$ instead of $\varphi(v; G)$ and $\mathcal{E}(v; G)$, respectively if G is obvious from the context.

3 Discovery Method

3.1 Super-mediator

As mentioned in section 1, we address the problem of discovering a different kind of influential nodes, which we call "super-mediator", i.e. those nodes which play an important role to pass the information to other nodes. Figure 1 (a) shows an example where it is suggested that there exist such super-mediator nodes. In this figure, by independently performing simulations $5,000$ times based on the IC model, we plotted $5,000$ curves for influence degree from some information source node with respect to time steps[3]. From this figure, we can observe that 1) due to its stochastic nature, each diffusion sample varies in a quite wide range for each simulation; and 2) some curves clearly exhibit sigmoidal behavior in part, in each of which the influence degree suddenly becomes relatively high during a certain time interval. From the latter observation, we can conjecture that there exist some super-mediator nodes which play an important role to pass the information to other nodes.

[3] The network used to generate these data is the blog network (see subsection 4.1).



(a) Diffusion samples (b) Influence degree distribution

Fig. 1: Information diffusion from some node in the blog network for the IC model ($p = 0.1$).

In Figure 1 (b), we plotted the distribution of the final influence degree for the above $5,000$ simulations. From this figure, we can observe that there exist a number of bell-shaped curves (which can be approximated by quadratic equations) in a logarithmic scale for each axis, which suggests that the influence degree distribution consists of several lognormal like distributions. When combining the observation from Figure 1 (a), we conjecture that super-mediators appear as a limited number of active nodes in some lognormal components with relatively high influence degree. Therefore, in order to discover these super-mediator nodes from information diffusion samples, we attempt to divide the diffusion samples in two groups (lower and upper), each assuming some probability distribution, find the best split by maximizing the likelihood, and rank the nodes in the upper samples by the F-measure.

3.2 Clustering of Diffusion samples

Let $\mathcal{S}(v) = \{1, 2, \cdots, M(v)\}$ denote a set of indices with respect to information diffusion samples for an information source node v, i.e. $\{d_1(v), d_2(v), \cdots, d_{M(v)}(v)\}$. Here note that $d_m(v)$ stands for a set of active nodes in the m-th diffusion sample. As described earlier, in order to discover super-mediator nodes, we consider dividing $\mathcal{S}(v)$ into two groups, $\mathcal{S}_1(v)$ and $\mathcal{S}_2(v)$, which are the upper group of samples with relatively high influence degree and the lower group, respectively. Namely, $\mathcal{S}_1(v) \cup \mathcal{S}_2(v) = \mathcal{S}(v)$ and $\min_{m \in \mathcal{S}_1(v)} |d_m(v)| > \max_{m \in \mathcal{S}_2(v)} |d_m(v)|$. Although we can straightforwardly extend our approach in case of k-groups division, we focus ourselves on the simplest case ($k = 2$) because of ease of both evaluation of basic performance and the following derivation. By assuming the independence of each sample drawn from either the upper or the lower group, we can consider the following likelihood function.

$$\mathcal{L}(\mathcal{S}(v); \mathcal{S}_1(v), \Theta) = \prod_{k \in \{1,2\}} \prod_{m \in \mathcal{S}_k(v)} p(m; \theta_k), \tag{1}$$

where $p(m; \theta_k)$ denotes some probability distribution with the parameter set θ_k for the m-th diffusion sample, and $\Theta = \{\theta_1, \theta_2\}$. If it is assumed that the influence degree distri-

bution consists of lognormal components, we can express $p(m;\theta_k)$ by

$$p(m;\theta_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}|d_m(v)|} \exp\left(-\frac{(\log|d_m(v)| - \mu_k)^2}{2\sigma_k^2}\right), \tag{2}$$

where $\theta_k = \{\mu_k, \sigma_k^2\}$. Then, based on the maximum likelihood estimation, we can identify the optimal upper group $\hat{\mathcal{S}}_1(v)$ by the following equation.

$$\hat{\mathcal{S}}_1(v) = \arg\max_{\mathcal{S}_1(v)} \left\{\mathcal{L}(\mathcal{S}; \mathcal{S}_1(v), \hat{\Theta})\right\}, \tag{3}$$

where $\hat{\Theta}$ denotes the set of maximum likelihood estimators.

Below we describe our method for efficiently obtaining $\hat{\mathcal{S}}_1(v)$ by focusing on the case that $p(m;\theta_k)$ is the lognormal distribution defined in Equation (2), although the applicability of the method is not limited to this case. For a candidate upper group $\mathcal{S}_1(v)$, by noting the following equations of the maximum likelihood estimation,

$$\hat{\mu_k} = \frac{1}{|\mathcal{S}_k(v)|} \sum_{m\in\mathcal{S}_k(v)} \log|d_m(v)|, \quad \hat{\sigma_k^2} = \frac{1}{|\mathcal{S}_k(v)|} \sum_{m\in\mathcal{S}_k(v)} (\log|d_m(v)| - \hat{\mu_k})^2, \tag{4}$$

we can transform Equation (3) as follows.

$$\hat{\mathcal{S}}_1(v) = \arg\max_{\mathcal{S}_1(v)} \left\{2\log\mathcal{L}(\mathcal{S}(v); \mathcal{S}_1(v), \hat{\Theta})\right\} = \arg\max_{\mathcal{S}_1(v)} \left\{-\sum_{k\in\{1,2\}} |\mathcal{S}_k| \log\left(\hat{\sigma_k^2}\right)\right\}. \tag{5}$$

Therefore, when a candidate upper group $\mathcal{S}_1(v)$ is successively changed by shifting its boundary between $\mathcal{S}_1(v)$ and $\mathcal{S}_2(v)$, we can efficiently obtain $\hat{\mathcal{S}}_1(v)$ by simply updating the sufficient statistics for calculating the maximum likelihood estimators. Here, we define the following operation to obtain the set of elements with the maximum influence degree,

$$\eta(\mathcal{S}(v)) = \left\{m; |d_m(v)| = \max_{m\in\mathcal{S}(v)} \{|d_m(v)|\}\right\}, \tag{6}$$

because there might exist more than one diffusion sample with the same influence degree. Then, we can summarize our algorithm as follows.

1. Initialize $\mathcal{S}_1(v) \leftarrow \eta(\mathcal{S}(v))$, $\mathcal{S}_2(v) \leftarrow \mathcal{S}(v) \setminus \eta(\mathcal{S}(v))$, and $\hat{L} \leftarrow -\infty$.
2. Iterate the following procedure:
2-1. Set $\mathcal{S}_1(v) \leftarrow \mathcal{S}_1(v) \cup \eta(\mathcal{S}_2(v))$, and $\mathcal{S}_2(v) \leftarrow \mathcal{S}_2(v) \setminus \eta(\mathcal{S}_2(v))$.
2-2. If $\mathcal{S}_2(v) = \eta(\mathcal{S}_2(v))$, then terminate the iteration.
2-3. Calculate $L = -\sum_{k\in\{1,2\}} |\mathcal{S}_k(v)| \log(\hat{\sigma_k^2})$.
2-4. If $\hat{L} < L$ then set $\hat{L} \leftarrow L$ and $\hat{\mathcal{S}}_1(v) \leftarrow \mathcal{S}_1(v)$
3. Output $\hat{\mathcal{S}}_1(v)$, and terminate the algorithm.

We describe the computational complexity of the above algorithm. Clearly, the number of iterations performed in step 2 is at most $(M(v) - 2)$. On the other hand, when applying the operator $\eta(\cdot)$ in steps 1 and 2.1 (or 2.2), by classifying each diffusion

sample according to its influence degree in advance, we can perform these operations with computational complexity of $O(1)$. Here note that since the influence degree is a positive integer less than or equal to $|V|$, we can perform the classification with computational complexity of $O(M(v))$. As for step 2.3, by adding (or removing) statistics calculated from $\eta(\mathcal{S}_2(v))$, we can update the maximum likelihood estimators $\hat{\Theta}$ defined in Equation (4) with computational complexity of $O(1)$. Therefore, the total computational complexity of our clustering algorithm is $O(M(v))$. Note that the above discussion can be applicable to a more general case for which the sufficient statistics of $p(m;\theta_k)$ is available to its parameter estimation.

A standard approach to the above clustering problem might be applying the EM algorithm by assuming a mixture of lognormal components. However, this approach is likely to confront the following drawbacks: 1) due to the local optimal problem, a number of parameter estimation trials are generally required by changing the initial parameter values, and we cannot guarantee the global optimality for the final result; 2) since many iterations are required for each parameter estimation trial, we need a substantially large computational load for obtaining the solution, which results in a prohibitively large processing time especially for a large data set; and 3) in case that a data set contains malicious outlier samples, we need a special care to avoid some unexpected problems such as degradation of $\hat{\sigma_k^2}$ to 0. Actually, our preliminary experiments based on this approach suffered from these drawbacks. In contrast, our proposed method always produces the optimal result with computational complexity of $O(M(v))$.

3.3 Super-mediator Discovery

Next, we describe our method for discovering super-mediator nodes. Let $D = \{d_m(v); v \in V, m = 1, \cdots, M(v)\}$ denote a set of observed diffusion samples. By using the above clustering method, we can estimate the upper group $\hat{\mathcal{S}}_1$ for each node $v \in V$. For $\hat{\mathcal{S}}_1(v)$, we employ, as a natural super-mediator score for a node $w \in V$, the following F-measure $F(w;v)$, a widely used measure in information retrieval, which is the harmonic average of recall and precision of a node w for the node v. Here the the recall means the number of samples that include the node w in the upper group divided by the total number of samples in the upper group, and the precision means the number of samples that include a node w in the upper group divided by the total number of the node w in the samples.

$$F(w;v) = \frac{2|\{m; m \in \hat{\mathcal{S}}_1(v), w \in d_m(v)\}|}{|\hat{\mathcal{S}}_1(v)| + |\{m; m \in \mathcal{S}(v), w \in d_m(v)\}|}. \tag{7}$$

Note that instead of the F-measure, we can employ the other measures such as the Jaccard coefficients, but for our objective that discovers characteristic nodes appearing in $\hat{\mathcal{S}}_1(v)$, we believe that the F-measure is most basic and natural. Then, we can consider the following expected F-measure for D.

$$\mathcal{F}(w) = \sum_{v \in V} F(w;v) r(v), \tag{8}$$

where $r(v)$ stands for the probability that the node v becomes an information source node, which can be empirically estimated by $r(v) = M(v)/\sum_{v \in V} M(v)$. Therefore, we

can discover candidates for the super-mediator nodes by ranking the nodes according to the above expected F-measure.

In order to confirm the validity of the F-measure and characterize its usefulness, we compare the ranking by the F-measure with the rankings by two other measures, and investigate how these rankings are different from or correlated to each other considering several situations. The first one is the expected influence degree defined in Section 2.3. From observed diffusion samples D, we can estimate it as follows.

$$\mathcal{E}(w) = \frac{1}{M(w)} \sum_{m=1}^{M(w)} |d_m(w)|. \tag{9}$$

The second one is the following measure:

$$\mathcal{N}(w) = \sum_{v \in V} |\{m; w \in d_m(v)\}| r(v). \tag{10}$$

This measure ranks high those nodes that are easily influenced by many other nodes.

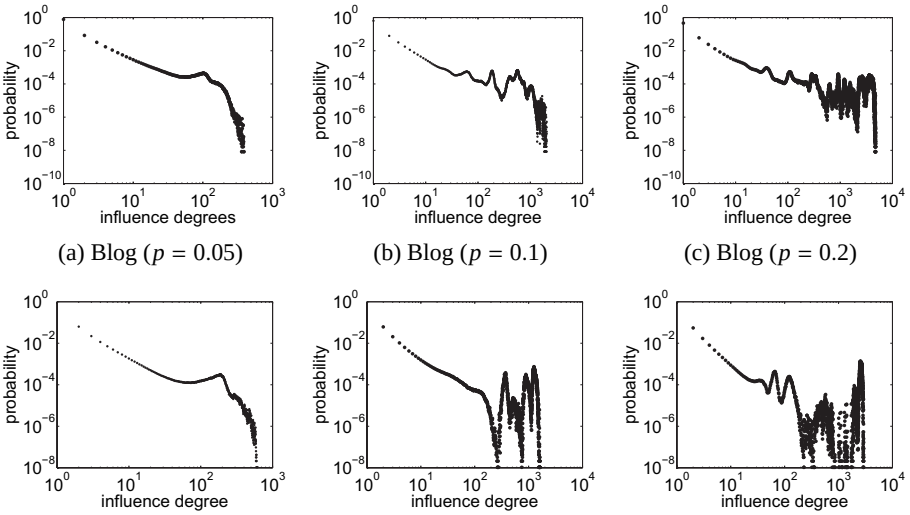
4 Experimental Evaluation

4.1 Data Sets

We employed two datasets of large real networks, which are both bidirectionally connected networks. The first one is a trackback network of Japanese blogs used in [13] and has $12,047$ nodes and $79,920$ directed links (the blog network). The other one is a network of people derived from the "list of people" within Japanese Wikipedia, also used in [13], and has $9,481$ nodes and $245,044$ directed links (the Wikipedia network).

Here, according to [17], we assumed the simplest case where the parameter values are uniform across all links and nodes, i.e. $p_{u,v} = p$ for the IC model. As for the LT model, we assumed $\omega_{u,v} = q|B(v)|^{-1}$, and adopted q ($0 \leq q \leq 1$) as the unique parameter for a network instead of $\omega_{u,v}$ as in [18]. According to [8], we set p to a value smaller than $1/\bar{d}$, where $\bar{d}$ is the mean out-degree of a network. Thus, the value of p was set to 0.1 for the blog network and 0.02 for the Wikipedia network. These are the base values, but in addition to them, we used two other values, one two times larger and the other two times smaller for our analyses, i.e. 0.02 and 0.05 for the blog network, and 0.04 and 0.01 for the Wikipedia network. We set the base value for q to be 0.9 for the both networks to achieve reasonably long diffusion results. Same as p, we also adopted two other values, one two times larger and the other two times smaller. Since the double of 0.9 exceeds the upper-bound of q, i.e. 1.0, we used 1.0 for the larger value, and we used 0.45 for the smaller one.

For each combination of these values, information diffusion samples were generated for the corresponding model on each network using each node in the network as the initial active node. In our experiments, we set $M = 10,000$, which means $10,000$ information diffusion samples were generated for each initial active node. Then, we analyzed them to discover super-mediators. To efficiently generate those information diffusion samples and estimate the expected influence degree $\mathcal{E}$ of an initial active node,



(a) Blog ($p = 0.05$) (b) Blog ($p = 0.1$) (c) Blog ($p = 0.2$)

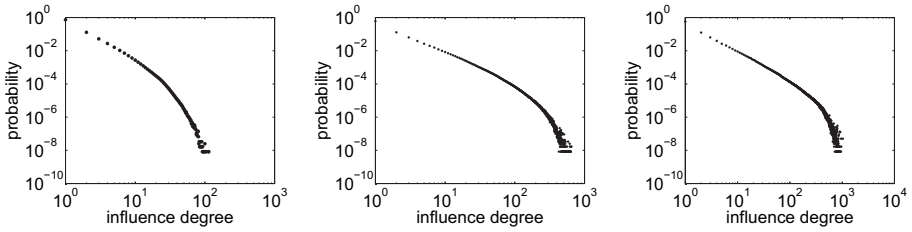
(d) Wikipedia ($p = 0.01$) (e) Wikipedia ($p = 0.02$) (f) Wikipedia ($p = 0.04$)

Fig. 2: The average influence degree distribution of the IC model

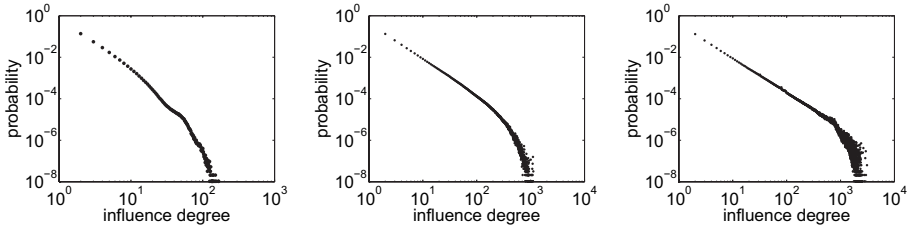
we adopted the method based on the bond percolation proposed in [14]. Note that we only use these two networks to generate the diffusion sample data which we assume we observed. Once the data are obtained, we no more use the network structure.

4.2 Influence Degree Distribution

First, we show the influence degree distribution for all nodes. Figure 2 is the results of the IC model and Fig. 3 is the results of the LT model. $M(= 10,000)$ simulations were performed for each initial node $v \in V$ and this is repeated for all the nodes in the network. Since the number of the nodes $|V|$ is about 10,000 for both the blog and the Wikipedia networks, these results are computed from about one hundred million diffusion samples and exhibits global characteristics of the distribution. We see that the distribution of the IC model consists of lognormal like distributions for a wide range of diffusion probability p with clearer indication for a larger p. Here it is known that if the variance of the lognormal distribution is large, it can be reasonably approximated by a power-law distribution [23]. On the contrary, we note that the distribution of the LT model is different and is a monotonically decreasing power-law like distribution. This observation is almost true of the distribution for an individual node v except that the distribution has one peak for the LT model. One example is already shown in Fig 1 (b) for the IC model. Figures 4 and 5 show some other results for the both models. In each of these figures the most influential node for the parameter used was chosen as the initial activated source node v. From this observation, the discovery model we derived in subsections 3.2 and 3.3 can be straightforwardly applied to the IC model by assuming that the probability distribution consists of lognormal components and the succeeding experiments were performed for the IC model. However, this does not necessarily

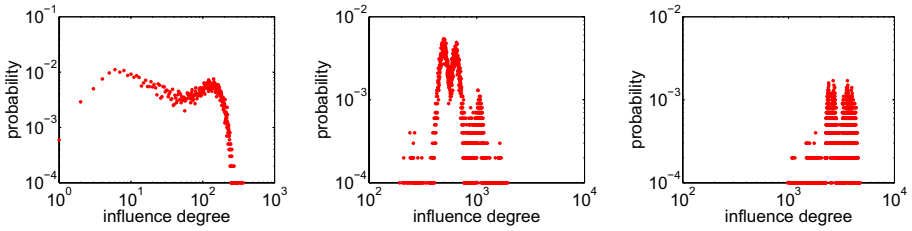


(a) Blog (q = 0.45) (b) Blog (q = 0.9) (c) Blog (q = 1.0)

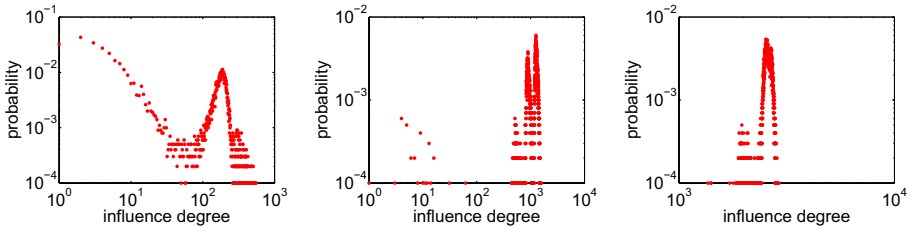


(d) Wikipedia (q = 0.45) (e) Wikipedia (q = 0.9) (f) Wikipedia (q = 1.0)

Fig. 3: The average influence degree distribution of the LT model



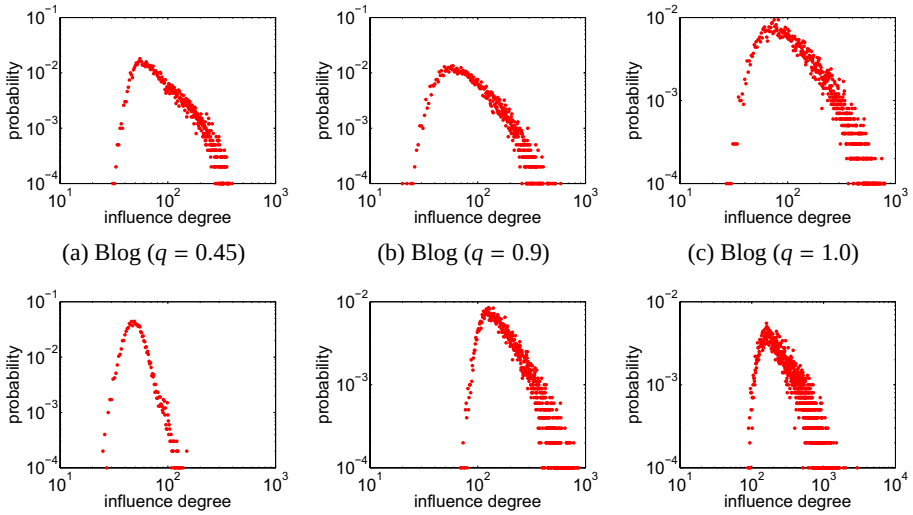
(a) Blog (p = 0.05) (b) Blog (p = 0.1) (c) Blog (p = 0.2)



(d) Wikipedia (p = 0.01) (e) Wikipedia (p = 0.02) (f) Wikipedia (p = 0.04)

Fig. 4: The influence degree distribution for a specific node v of the IC model

mean that the notion of super-mediator is only applicable to the IC model. Finding a reasonable and efficient way to discover super-mediator nodes for the LT model is our on-going research topic. Further, the assumption of dividing the groups into only two need be justified. This is also left to our future work.



(a) Blog ($q = 0.45$) (b) Blog ($q = 0.9$) (c) Blog ($q = 1.0$)

(d) Wikipedia ($q = 0.45$) (e) Wikipedia ($q = 0.9$) (f) Wikipedia ($q = 1.0$)

Fig. 5: The influence degree distribution for a specific node v of the LT model

4.3 Super-mediator Ranking

Tables 1, 2 and 3 summarize the ranking results. Ranking is evaluated for two different values of diffusion probability ($p = 0.1$ and $p = 0.05$ for the blog data, and $p = 0.02$ and 0.01 for the Wikipedia data) and for the three measures mentioned in subsection 3.3. Rank by all the measures is based on the value rounded off to three decimal places. So the same rank appears more than once. The first two (Tables 1 and 2) rank the nodes by $\mathcal{F}$ for $p = 0.1$ and 0.05 (blog data) and $p = 0.02$ and 0.01 (Wikipedia data), respectively,

Table 1: Comparison of the ranking by $\mathcal{F}$ with rankings by $\mathcal{E}$ and $\mathcal{N}$ for a large diffusion probability.

(a) Blog network ($p = 0.1$)

Ranking by $\mathcal{F}$		Ranking by $\mathcal{E}/\mathcal{N}$	
Ranking	Node ID	$\mathcal{E}$	$\mathcal{N}$
1	146	2	2
1	155	1	1
3	140	3	3
3	150	4	4
5	238	5	5
5	278	6	6
5	240	7	7
5	618	10	8
9	136	8	9
9	103	9	10

(b) Wikipedia network ($p = 0.02$)

Ranking by $\mathcal{F}$		Ranking by $\mathcal{E}/\mathcal{N}$	
Ranking	Node ID	$\mathcal{E}$	$\mathcal{N}$
1	790	1	1
1	8340	2	2
3	323	3	3
3	279	4	4
5	326	5	5
6	772	6	6
6	325	7	7
8	1407	8	8
9	4924	9	9
10	3149	11	10

Table 2: Comparison of the ranking by $\mathcal{F}$ with rankings by $\mathcal{E}$ and $\mathcal{N}$ for a small diffusion probability.

(a) Blog network ($p = 0.05$)

Ranking by $\mathcal{F}$		Ranking by $\mathcal{E}/\mathcal{N}$	
Ranking	Node ID	$\mathcal{E}$	$\mathcal{N}$
1	155	26	28
2	146	29	29
3	140	41	44
4	150	63	66
5	238	92	93
6	618	79	81
6	240	113	112
8	103	84	86
8	490	95	96
8	173	88	89

(b) Wikipedia network ($p = 0.01$)

Ranking by $\mathcal{F}$		Ranking by $\mathcal{E}/\mathcal{N}$	
Ranking	Node ID	$\mathcal{E}$	$\mathcal{N}$
1	790	167	168
2	279	199	198
2	4019	1	1
4	3729	2	2
4	7919	3	3
4	1720	7	4
4	4465	5	6
4	1712	6	7
9	4380	4	5
9	3670	9	8

Table 3: Comparison of the ranking by $\mathcal{E}$ for a high diffusion probability with rankings by $\mathcal{E}$, $\mathcal{F}$, and $\mathcal{N}$ for a low diffusion probability.

(a) Blog network

Ranking by $\mathcal{E}$ for $p = 0.1$		Ranking by $\mathcal{E}/\mathcal{F}/\mathcal{N}$ for $p = 0.05$		
Ranking	Node ID	$\mathcal{E}$	$\mathcal{F}$	$\mathcal{N}$
1	155	26	1	28
2	146	29	2	29
3	140	41	3	44
4	150	63	4	66
5	238	92	5	93
6	278	161	18	154
7	240	113	6	112
8	136	83	8	85
9	103	84	8	86
10	618	79	6	81

(b) Wikipedia network

Ranking by $\mathcal{E}$ for $p = 0.02$		Ranking by $\mathcal{E}/\mathcal{F}/\mathcal{N}$ for $p = 0.01$		
Ranking	Node ID	$\mathcal{E}$	$\mathcal{F}$	$\mathcal{N}$
1	790	167	1	168
2	8340	200	9	201
3	323	196	14	200
4	279	199	2	198
5	326	212	24	206
6	325	231	51	236
7	772	242	41	235
8	1407	257	80	264
9	4924	305	111	298
10	2441	279	103	287

and compare each ranking with those by $\mathcal{E}$ and $\mathcal{N}$. From these results we observe that when the diffusion probability is large all the three measures ranks the nodes in a similar way. This means that the influential nodes also play a role of super-mediator for the other source nodes. When the diffusion probability is small, the Wikipedia data still shows the similar tendency but the blog data does not. We further note that $\mathcal{E}$ and $\mathcal{N}$ rank the nodes in a similar way regardless of the value of diffusion probability. This is understandable because the both networks are bidirectional. In summary, when the diffusion probability is large, all the three measures are similar and the influential nodes also play a role of super-mediator for the other source nodes.

The third one (Table 3) ranks the nodes by $\mathcal{E}$ for $p = 0.01$ (blog data) and $p = 0.02$ (Wikipedia data) and compares them with the rankings by the three measures for $p = 0.05$ (blog data) and $p = 0.01$ (Wikipedia data). The results say that the influ-

ential nodes are different between the two different diffusion probabilities, but what is strikingly interesting to note is that the nodes that are identified to be influential (up to 10th) at large diffusion probability are almost the same as the nodes that rank high by $\mathcal{F}$ at small diffusion probability for the blog data. This correspondence is not that clear for the Wikipedia data but the correlation of the rankings by $\mathcal{E}$ (at large diffusion probability) and $\mathcal{F}$ (at small diffusion probability) is much larger than the corresponding correlation by the other two measures ($\mathcal{E}$ and $\mathcal{N}$). This implies that the super-mediators at small diffusion probability become influential at large diffusion probability. Since the F-measure can be evaluated by the observed information sample data alone and there is no need to know the network structure, this fact can be used to predict which nodes become influential when the diffusion probability switches from a small value for which we have enough data to a large value for which we do not have any data yet.

4.4 Characterization of Super-mediator and Discussions

If we observe that some measure evaluated for a particular value of diffusion probability gives an indication of the influential nodes when the value of diffusion probability is changed, it would be a useful measure for finding influential nodes for a new situation. It is particularly useful when we have abundant observed set of information diffusion samples with normal diffusion probability and we want to discover high ranked influential nodes in a case where the diffusion probability is larger. For example, this problem setting corresponds to predicting the influential nodes for the unexperienced rapid spread of new information, e.g. spread of new acute contagion, because it is natural to think that we have abundant data for the spread of normal moderate contagion.

The measure based on $\mathcal{E}$ ranks high those nodes that are also influential where the diffusion probability is different from the current value if nodes are not sensitive to the diffusion probability, i.e. a measure useful to estimate influential nodes from the known results when the diffusion probability changes under such a condition. The measure based on $\mathcal{N}$ ranks high those nodes that are easily influenced by many other nodes. It is a measure useful to estimate influential nodes from the known results if they are the nodes easily influenced by other nodes. In our experiments, the influential nodes by E for the much larger diffusion probability, i.e. $p = 0.2$ (blog data) and $p = 0.04$ (Wikipedia data) were almost the same as the high ranked ones by any one of the three measures E, N and F for $p = 0.1$ (blog data) and $p = 0.02$ (Wikipedia data), although we have to omit the details due to the space limitation.

In the previous subsection we showed that the super-mediators at small diffusion probability become influential at large diffusion probability. In a situation where there are relatively large number of active nodes, the probability that more than one parent try to activate their same child increases, which mirrors the situation where the diffusion probability is effectively large. It is the super-mediators that play the central role in these active node group under such a situation. This would explain why the super-mediators at the small diffusion probability become influential nodes at the large diffusion probability.

5 Conclusion

We found that the influence degree for the IC model exhibits a distribution which is a mixture of two distributions (power-law like distribution and lognormal like distribution). This implied that there are nodes that may play different roles in information diffusion process. We made a hypothesis that there should be nodes that play an important role to pass the information to other nodes, and called these nodes "super-mediators". These nodes are different from what is usually called "influential nodes" (nodes that spread information as much as possible). We devised an algorithm based on maximum likelihood and linear search which can efficiently identify the super-mediator node group from the observed diffusion sample data, and proposed a measure based on recall and precision to rank the super-mediators. We tested our hypothesis by applying it to the information diffusion sample data generated by two real networks. We found that the high ranked super-mediators are also the high ranked influential nodes when the diffusion probability is large, i.e. the influential nodes also play a role of super-mediator for the other source nodes, but not necessarily so when the diffusion probability is small, and further, to our surprise, that when the high ranked super-mediators are different from the top ranked influential nodes, which is the case when the diffusion probability is small, those super-mediators become the high ranked influential nodes when the diffusion probability becomes larger. This finding will be useful to predict the influential nodes for the unexperienced spread of new information from the known experience, e.g. prediction of influential nodes for the spread of new acute contagion for which we have no available data yet from the abundant data we already have for the spread of moderate contagion.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters **12** (2001) 211–223
2. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Physical Review E **66** (2002) 035101
3. Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45** (2003) 167–256
4. Watts, D.J.: A simple model of global cascades on random networks. Proceedings of National Academy of Science, USA **99** (2002) 5766–5771
5. Watts, D.J., Dodds, P.S.: Influence, networks, and public opinion formation. Journal of Consumer Research **34** (2007) 441–458
6. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. SIGKDD Explorations **6** (2004) 43–52

7. Domingos, P.: Mining social networks for viral marketing. IEEE Intelligent Systems **20** (2005) 80–82
8. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003). (2003) 137–146
9. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06). (2006) 228–237
10. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07). (2007) 1371–1376
11. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Extracting influential nodes on a social network for information diffusion. Data Mining and Knowledge Discovery, Springer **20** (2010) 70–97
12. Kimura, M., Saito, K., Motoda, H.: Minimizing the spread of contamination by blocking links in a network. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08). (2008) 1175–1180
13. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data **3** (2009) 9:1–9:23
14. Kimura, M., Saito, K., Motoda, H.: Efficient estimation of influence functions fot sis model on social networks. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09). (2009)
15. Saito, K., Kimura, M., Motoda, H.: Discovering influential nodes for sis models in social networks. In: Proceedings of the Twelfth International Conference of Discovery Science (DS2009), Springer, LNAI 5808 (2009) 302–316
16. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Finding influential nodes in a social network from information diffusion data. In: Proceedings of the International Workshop on Social Computing and Behavioral Modeling (SBP09). (2009) 138–145
17. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Learning continuous-time information diffusion model for social behavioral data analysis. In: Proceedings of the 1st Asian Conference on Machine Learning (ACML2009). (2009) 322–337
18. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Behavioral analyses of information diffusion models by observed data of social network. In: Proceedings of the International Workshop on Social Computing and Behavioral Modeling (SBP10). (2010) 149–158
19. Goyal, A., Bonchi, F., Lakshhmanan, L.V.S.: Learning influence probabilities in social networks. In: Proceedings of the third ACM international conference on Web Search and Data Mining. (2010) 241–250
20. Bakshy, E., Karrer, B., Adamic, L.A.: Social influence and the diffusion of user-created content. In: Proceedings of the tenth ACM conference on Electronic Commerce. (2009) 325–334
21. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007). (2007) 420–429
22. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2009). (2009) 199–208
23. Mitzenmacher, M.: A brief history of generative models for power law and lognormal distributions. Internet Mathematics **1** (2004) 226–251

Selecting Information Diffusion Models over Social Networks for Behavioral Analysis

Kazumi Saito[1], Masahiro Kimura[2], Kouzou Ohara[3], and Hiroshi Motoda[4]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[3] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We investigate how well different information diffusion models can explain observation data by learning their parameters and discuss which model is better suited to which topic. We use two models (AsIC, AsLT), each of which is an extension of the well known Independent Cascade (IC) and Linear Threshold (LT) models and incorporates asynchronous time delay. The model parameters are learned by maximizing the likelihood of observation, and the model selection is performed by choosing the one with better predictive accuracy. We first show by using four real networks that the proposed learning algorithm correctly learns the model parameters both accurately and stably, and the proposed selection method identifies the correct diffusion model from which the data are generated. We next apply these methods to behavioral analysis of topic propagation using the real blog propagation data, and show that although the relative propagation speed of topics that are derived from the learned parameter values is rather insensitive to the model selected, there is a clear indication as to which topic better follows which model. The correspondence between the topic and the model selected is well interpretable.

1 Introduction

The growth of Internet has enabled to form various kinds of large-scale social networks, through which a variety of information including innovation, hot topics and even malicious rumors can be propagated in the form of so-called "word-of-mouth" communications. Social networks are now recognized as an important medium for the spread of information, and a considerable number of studies have been made [1–5]. Widely used information diffusion models in these studies are the *independent cascade (IC)* [6–8] and the *linear threshold (LT)* [9, 10]. They have been used to solve such problems as the *influence maximization problem* [7, 11].

These two models focus on different information diffusion aspects. The IC model is sender-centered and each active node *independently* influences its inactive neighbors with given diffusion probabilities. The LT model is receiver-centered and a node is influenced by its active neighbors if their total weight exceeds the threshold for the node. Which model is more appropriate depends on the situation and selecting the appropriate one is not easy. First of all, we need to know how different model behaves differently and how well or badly explain the observation data. Both models have parameters that need be specified in advance: diffusion probabilities for the IC model, and weights for the LT model. However, their true values are not known in practice. This poses yet another problem of estimating them from a set of information diffusion results that are observed as time-sequences of influenced (activated) nodes.

This falls in a well defined parameter estimation problem in machine learning framework. Given a generative model with some parameters and the observed data, it is possible to calculate the liklihood that the data are generated and the parameters can be estimated by maximizing the liklihood. This approach has a thorough theoretical background. In general, the way the parameters are estimated dependes on how the generative model is given. To the best of our knowledge, we are the first to follow this line of research. We addressed this problem for the IC model [12] and its varient that incorporates asynchronous time delay (referred to as the AsIC model) [13]. Gruhl et.al. also challenged the same problem of estimating the parameters and proposed an EM-like algorithm, but they did not formalize the liklihood and it is not clear what is being optimized in deriving the parameter update formulae. Goyal et.al attacked this problem from a different angle [14]. They employed a varient of the LT model and estimated the parameter values by four different methods, all of which are directly computed from the frequency of the events in the observed data. Their approach is efficient, but it is more likely ad hoc and lacks in theoretical evidence. Bakshy et.al [15] addressed the problem of diffusion of user-created content (asset) and used the maximum likelihood method to estimate the rate of asset adoption. However, they only modeled the rate of adoption and did not consider the diffusion model itself. Their focus is on data analysis.

In this paper, we first propose a method of learning the parameter values of a variant of the LT model that incorporates asynchronous time delay, similarly to the AsIC model, under the maximum liklihood framework. We refer to this diffusion model as the AsLT model. The model is similar to the one used in [14] but different in that we explicity model the delay of node activation after the activation condition has been satisfied. Next we propose a method of model selection based on the predictive accuracy, using the two models: AsIC and AsLT.

It is indispensable to be able to cope with asynchronous time delay to do realistic analyses of information diffusion because, in the real world, information propagates along the continuous time axis, and time-delays can occur during the propagation asynchronously. In fact, the time stamps of the observed data are not equally spaced. Thus, the proposed learning method has to estimate not only the weight parameters but also the time-delay parameters from the observed data. Incorporating time-delay makes the time-sequence observation data structural, which makes the analyes of diffusion process difficut because there is no way of knowing which node has activated which other node from the observation data sequence. Knowing the optimal parameter values does

not mean that the observation follows the model. We have to decide which model better explains the observation. We solve this problem by comparing the predictive accuracy of each model. We use a variant of hold-out method applied to a set of sequential data, which is similar to the leave-one-out method applied to a multiple time sequence data. Extensive experimens have been performed to evaluate the effectiveness of the proposed method using both artificially generated data and real observation data. Experiments that used artificial data using four real network structures showed that the method can correctly 1) learn the parameters and 2) select the model by which the data have been generated. Experiments that used real diffusion data of topic propagation showed that 1) both AsIC and AsLT models well capture the global characteristics of topic propagations but 2) the predictive accuracy of each model is different for each topic and some topics have clear indication as to which model each better follows.

2 Information Diffusion Models

We first present the *asynchronous independent cascade (AsIC) model* introduced in [13], and then define the *asynchronous linear threshold (AsLT) model*. We mathematically model the spread of information through a directed network $G = (V, E)$ without self-links, where V and E ($\subset V \times V$) stand for the sets of all the nodes and links, respectively. For each node v in the network G, we denote $F(v)$ as a set of child nodes of v, *i.e.*, $F(v) = \{w; (v, w) \in E\}$. Similarly, we denote $B(v)$ as a set of parent nodes of v, *i.e.*, $B(v) = \{u; (u, v) \in E\}$. We call nodes *active* if they have been influenced with the information. In the following models, we assume that nodes can switch their states only from inactive to active, but not the other way around, and that, given an initial active node set S, only the nodes in S are active at an initial time.

2.1 Asynchronous Independent Cascade Model

We first recall the definition of the IC model according to [7], and then introduce the AsIC model. In the IC model, we specify a real value $\kappa_{u,v}$ with $0 < \kappa_{u,v} < 1$ for each link (u, v) in advance. Here $\kappa_{u,v}$ is referred to as the *diffusion probability* through link (u, v). The diffusion process unfolds in discrete time-steps $t \geq 0$, and proceeds from a given initial active set S in the following way. When a node u becomes active at time-step t, it is given a single chance to activate each currently inactive child node v, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time-step $t+1$. If multiple parent nodes of v become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

In the AsIC model, we specify real values $r_{u,v}$ with $r_{u,v} > 0$ in advance for each link $(u, v) \in E$ in addition to $\kappa_{u,v}$, where $r_{u,v}$ is referred to as the *time-delay parameter* through link (u, v). The diffusion process unfolds in continuous-time t, and proceeds from a given initial active set S in the following way. Suppose that a node u becomes active at time t. Then, u is given a single chance to activate each currently inactive child node v. We choose a delay-time δ from the exponential distribution with parameter $r_{u,v}$.

If v has not been activated before time $t + \delta$, then u attempts to activate v, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time $t + \delta$. Under the continuous time framework, it is unlikely that v is activated simultaneously by its multiple parent nodes exactly at time $t + \delta$. So we ignore this possibility. The process terminates if no more activations are possible.

2.2 Asynchronous Linear Threshold Model

Similarly to the above, we first define the LT model. In this model, for every node $v \in V$, we specify a *weight* ($\omega_{u,v} > 0$) from its parent node u in advance such that $\sum_{u \in B(v)} \omega_{u,v} \leq 1$. The diffusion process from a given initial active set S proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0, 1]$. At time-step t, an inactive node v is influenced by each of its active parent nodes, u, according to weight $\omega_{u,v}$. If the total weight from active parent nodes of v is no less than θ_v, that is, $\sum_{u \in B_t(v)} \omega_{u,v} \geq \theta_v$, then v will become active at time-step $t + 1$. Here, $B_t(v)$ stands for the set of all the parent nodes of v that are active at time-step t. The process terminates if no more activations are possible.

Next, we define the AsLT model. In the AsLT model, in addition to the weight set $\{\omega_{u,v}\}$, we specify real values r_v with $r_v > 0$ in advance for each node $v \in V$. We refer to r_v as the *time-delay parameter* on node v. Note that r_v depends only on v unlike $r_{u,v}$ of the AsIC model, which means that it is the node v's decision when to receive the information once the activation condition has been satisfied. The diffusion process unfolds in continuous-time t, and proceeds from a given initial active set S in the following way. Suppose that the total weight from active parent nodes of v became no less than θ_v at time t for the first time. Then, v will become active at time $t + \delta$, where we choose a delay-time δ from the exponential distribution with parameter r_v. Further, note that even if some other non-active parent nodes of v has become active during the time period between t and $t + \delta$, the activation time of v, $t + \delta$, still remains the same. The other diffusion mechanisms are the same as the LT model.

3 Learning Algorithms

We define the time-delay parameter vector $\boldsymbol{r}$ and the diffusion parameter vector $\boldsymbol{\kappa}$ by $\boldsymbol{r} = (r_{u,v})_{(u,v) \in E}$ and $\boldsymbol{\kappa} = (\kappa_{u,v})_{(u,v) \in E}$ for the AsIC model and the parameter vectors $\boldsymbol{\omega}$ and $\boldsymbol{r}$ by $\boldsymbol{\omega} = (\omega_{u,v})_{(u,v) \in E}$ and $\boldsymbol{r} = (r_v)_{v \in V}$ for the AsLT model. We next consider an observed data set of M independent information diffusion results, $\{D_m;\ m = 1, \cdots, M\}$. Here, each D_m is a set of pairs of active nodes and their activation times in the mth diffusion result, $D_m = \{(u, t_{m,u}),\ (v, t_{m,v}),\ \cdots\}$. For each D_m, we denote the observed initial time by $t_m = \min\{t_{m,v};\ (v, t_{m,v}) \in D_m\}$, and the observed final time by $T_m \geq \max\{t_{m,v};\ (v, t_{m,v}) \in D_m\}$. Note that T_m is not necessarily equal to the final activation time. Hereafter, we express our observation data by $\mathcal{D}_M = \{(D_m, T_m);\ m = 1, \cdots, M\}$. For any $t \in [t_m, T_m]$, we set $C_m(t) = \{v;\ (v, t_{m,v}) \in D_m,\ t_{m,v} < t\}$. Namely, $C_m(t)$ is the set of active nodes before time t in the mth diffusion result. For convenience sake, we use C_m as referring to the set of all the active nodes in the mth diffusion result. Moreover, we define a set of non-active nodes with at least one active parent node for each by

$\partial C_m = \{v;\ (u,v) \in E,\ u \in C_m,\ v \notin C_m\}$. For each node $v \in C_m \cup \partial C_m$, we define the following subset of parent nodes, each of which has a chance to activate v.

$$\mathcal{B}_{m,v} = \begin{cases} B(v) \cap C_m(t_{m,v}) & \text{if } v \in C_m(t_{m,v}), \\ B(v) \cap C_m & \text{if } v \in \partial C_m. \end{cases}$$

Note that the underlying model behind the observed data is not available in reality. Thus, we investigate how the model affects the information diffusion results, and consider selecting a model which better explains the given observed data from the candidates, i.e., AsIC and AsLT models. To this end, we first have to estimate the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ for the AsIC model, and the values of $\boldsymbol{r}$ and $\boldsymbol{\omega}$ for the AsLT model for the given $\mathcal{D}_M$. For the former, we adopt the method proposed in [13], which is only briefly explained here. For the latter, we propose a novel method of estimating those values.

3.1 Learning Parameters of AsIC Model

To estimate the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$ for the AsIC model, We derived the following likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ to use as the objective function [13],

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M) = \prod_{m=1}^{M} \prod_{v \in C_m} \left(h_{m,v} \prod_{w \in F(v) \setminus C_m} g_{m,v,w} \right), \tag{1}$$

where $h_{m,v}$ is the probability density that the node v such that $v \in D_m$ with $t_{m,v} > 0$ for the mth diffusion result is activated at time $t_{m,v}$, and $g_{m,v,w}$ is the probability that a node w is not activated by a node v within the observed time period $[t_m, T_m]$ when there is a link $(v,w) \in E$ and $v \in C_m$ for the mth diffusion result. Then, we derived an iterative algorithm to stably obtain the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ that maximize equation (1). Please refer to [13] for mode details. Hereafter, we refer to this method as the *AsIC model based method*.

3.2 Learning Parameters of AsLT Model

Likelihood function To estimate the values of $\boldsymbol{r}$ and $\boldsymbol{\omega}$ from $\mathcal{D}_M$ for the AsLT model, we first derive the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\omega}$ in a rigorous way to use as the objective function. For the sake of technical convenience, we introduce a slack weight $\omega_{v,v}$ for each node $v \in V$ such that $\omega_{v,v} + \sum_{u \in B(v)} \omega_{u,v} = 1$. Here note that we can regard each weight $\omega_{*,v}$ as a multinomial probability since a threshold θ_v is chosen uniformly at random from the interval $[0, 1]$ for each node v.

Suppose that a node v became active at time $t_{m,v}$ for the mth result. Then, we know that the total weight from active parent nodes of v became no less than θ_v at the time when one of them, $u \in \mathcal{B}_{m,v}$, became first active. However, in case of $|\mathcal{B}_{m,v}| > 1$, there is no way of exactly knowing the actual nodes due to the asynchronous time-delay. Suppose that a node v was actually activated when a node $\zeta \in \mathcal{B}_{m,v}$ became activated. Then θ_v is between $\sum_{u \in B(v) \cap C_m(t_{m,\zeta})} \omega_{u,v}$ and $\omega_{\zeta,v} + \sum_{u \in B(v) \cap C_m(t_{m,\zeta})} \omega_{u,v}$. Namely, the probability

that θ_v is chosen from this range is $\omega_{\zeta,v}$. Here note that such events with respect to different active parent nodes are mutually disjoint. Thus, the probability density that the node v is activated at time $t_{m,v}$, denoted by $h_{m,v}$, can be expressed as

$$h_{m,v} = \sum_{u \in \mathcal{B}_{m,v}} \omega_{u,v} r_v \exp(-r_v(t_{m,v} - t_{m,u})). \tag{2}$$

Here we define $h_{m,v} = 1$ if $t_{m,v} = t_m$.

Next, we consider any node $w \in V$ belonging to $\partial C_m = \{w; (v, w) \in E \wedge v \in C_m(T_m) \wedge w \notin C_m(T_m)\}$ for the mth result. Let $g_{m,v}$ denote the probability that the node v is not activated within the observed time period $[t_m, T_m]$. We can calculate $g_{m,v}$ as

$$\begin{aligned} g_{m,v} &= 1 - \sum_{u \in \mathcal{B}_{m,v}} \omega_{u,v} \int_{t_{m,u}}^{T_m} r_v \exp(-r_v(t - t_{m,u}))dt = 1 - \sum_{u \in \mathcal{B}_{m,v}} \omega_{u,v}(1 - \exp(-r_v(T_m - t_{m,u}))) \\ &= \omega_{v,v} + \sum_{u \in B(v) \setminus \mathcal{B}_{m,v}} \omega_{u,v} + \sum_{u \in \mathcal{B}_{m,v}} \omega_{u,v} \exp(-r_v(T_m - t_{m,u})). \end{aligned} \tag{3}$$

Therefore, by using Equations (2) and (3), and the independence properties, we can define the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\omega}$ by

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M) = \prod_{m=1}^{M} \left(\prod_{v \in C_m} h_{m,v} \right) \left(\prod_{v \in \partial C_m} g_{m,v} \right). \tag{4}$$

Thus, our problem is to obtain the time-delay parameter vector $\boldsymbol{r}$ and the diffusion parameter vector $\boldsymbol{\omega}$, which together maximize Equation (4).

Learning Algorithm For the above learning problem, we can derive an estimation method based on the Expectation-Maximization algorithm in order to stably obtain its solutions. Hereafter, we refer to this proposed method as the *AsLT model based method*. By the following formulas, we define $\phi_{m,u,v}$ for each $v \in C_m$ and $u \in \mathcal{B}_{m,v}$, $\varphi_{m,u,v}$ for each $v \in \partial C_m$ and $u \in \{v\} \cup B(v) \setminus \mathcal{B}_{m,v}$, and $\psi_{m,u,v}$ for each $v \in \partial C_m$ and $u \in \mathcal{B}_{m,v}$, respectively.

$$\begin{aligned} &\phi_{m,u,v} = \omega_{u,v} r_v \exp(-r_v(t_{m,v} - t_{m,u})) \,/\, h_{m,v}, \quad \varphi_{m,u,v} = \omega_{u,v} \,/\, g_{m,v}, \\ &\psi_{m,u,v} = \omega_{u,v} \exp(-r_v(T_m - t_{m,u})) \,/\, g_{m,v}. \end{aligned}$$

Let $\bar{\boldsymbol{r}} = (\bar{r}_v)$ and $\bar{\boldsymbol{\omega}} = (\bar{\omega}_{u,v})$ be the current estimates of $\boldsymbol{r}$ and $\boldsymbol{\omega}$, respectively. Similarly, let $\bar{\phi}_{m,u,v}$, $\bar{\varphi}_{m,u,v}$, and $\bar{\psi}_{m,u,v}$ denote the values of $\phi_{m,u,v}$, $\varphi_{m,u,v}$, and $\psi_{m,u,v}$ calculated by using $\bar{\boldsymbol{r}}$ and $\bar{\boldsymbol{\omega}}$, respectively.

From equations (2), (3), (4), we can transform $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M)$ as follows:

$$\log \mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M) = Q(\boldsymbol{r}, \boldsymbol{\omega}; \bar{\boldsymbol{r}}, \bar{\boldsymbol{\omega}}) - H(\boldsymbol{r}, \boldsymbol{\omega}; \bar{\boldsymbol{r}}, \bar{\boldsymbol{\omega}}), \tag{5}$$

where $Q(\boldsymbol{r}, \boldsymbol{\omega}; \bar{\boldsymbol{r}}, \bar{\boldsymbol{\omega}})$ is defined by

$$Q(\boldsymbol{r}, \boldsymbol{\omega}; \bar{\boldsymbol{r}}, \bar{\boldsymbol{\omega}}) = \sum_{m=1}^{M} \left(\sum_{v \in C_m} Q_{m,v}^{(1)} + \sum_{v \in \partial C_m} Q_{m,v}^{(2)} \right), \tag{6}$$

$$Q^{(1)}_{m,v} = \sum_{u\in\mathcal{B}_{m,v}} \bar{\phi}_{m,u,v} \log(\omega_{u,v} r_v \exp(-r_v(t_{m,v} - t_{m,u})))$$
$$Q^{(2)}_{m,v} = \sum_{u\in\{v\}\cup B(v)\setminus\mathcal{B}_{m,v}} \bar{\varphi}_{m,u,v} \log(\omega_{u,v}) + \sum_{u\in\mathcal{B}_{m,v}} \bar{\psi}_{m,u,v} \log(\omega_{u,v} \exp(-r_v(T_m - t_{m,u}))).$$

It is easy to see that $Q(\boldsymbol{r}, \boldsymbol{\omega}; \bar{\boldsymbol{r}}, \bar{\boldsymbol{\omega}})$ is convex with respect to $\boldsymbol{r} and \boldsymbol{\omega}$, and $H(\boldsymbol{r}, \boldsymbol{\kappa}; \bar{\boldsymbol{r}}, \bar{\boldsymbol{\omega}})$ is defined by

$$H(\boldsymbol{r}, \boldsymbol{\omega}; \bar{\boldsymbol{r}}, \bar{\boldsymbol{\omega}}) = \sum_{m=1}^{M} \left(\sum_{v\in C_m} H^{(1)}_{m,v} + \sum_{v\in\partial C_m} H^{(2)}_{m,v} \right), \tag{7}$$

$$H^{(1)}_{m,v} = \sum_{u\in\mathcal{B}_{m,v}} \bar{\phi}_{m,u,v} \log(\phi_{m,u,v}), \quad H^{(2)}_{m,v} = \sum_{u\in\{v\}\cup B(v)\setminus C_m} \bar{\varphi}_{m,u,v} \log(\varphi_{m,u,v}) + \sum_{u\in\mathcal{B}_{m,v}} \bar{\psi}_{m,u,v} \log(\psi_{m,u,v}).$$

Since $H(\boldsymbol{r}, \boldsymbol{\omega}; \bar{\boldsymbol{r}}, \bar{\boldsymbol{\omega}})$ is maximized at $\boldsymbol{r} = \bar{\boldsymbol{r}}$ and $\boldsymbol{\omega} = \bar{\boldsymbol{\omega}}$ from equation (7), we can increase the value of $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M)$ by maximizing $Q(\boldsymbol{r}, \boldsymbol{\omega}; \bar{\boldsymbol{r}}, \bar{\boldsymbol{\omega}})$ (see equation (5)).

Thus, we obtain the following update formulas of our estimation method as the solution which maximizes $Q(\boldsymbol{r}, \boldsymbol{\omega}; \bar{\boldsymbol{r}}, \bar{\boldsymbol{\omega}})$ with respect to $\boldsymbol{r}$:

$$r_v = \left(\sum_{m\in\mathcal{M}^{(1)}_v} \sum_{u\in\mathcal{B}_{m,v}} \bar{\phi}_{m,u,v} \right) \times \left(\sum_{m\in\mathcal{M}^{(1)}_v} \sum_{u\in\mathcal{B}_{m,v}} \bar{\phi}_{m,u,v}(t_{m,v} - t_{m,u}) + \sum_{m\in\mathcal{M}^{(2)}_v} \sum_{u\in\mathcal{B}_{m,v}} \bar{\psi}_{m,u,v}(T_m - t_{m,u}) \right)^{-1}$$

where $\mathcal{M}^{(1)}_v$ and $\mathcal{M}^{(2)}_v$ are defined by

$$\mathcal{M}^{(1)}_v = \{m \in \{1, \cdots, M\};\ v \in C_m\}, \quad \mathcal{M}^{(2)}_v = \{m \in \{1, \cdots, M\};\ v \in \partial C_m\}.$$

As for $\boldsymbol{\omega}$, we have to take the constraints $\omega_{v,v} + \sum_{u\in B(v)} \omega_{u,v} = 1$ into account for each v, which can easily be made using the Lagrange multipliers method, and we obtain the following update formulas of our estimation method:

$$\omega_{u,v} \propto \sum_{m\in\mathcal{M}^{(1)}_{u,v}} \bar{\phi}_{m,u,v} + \sum_{m\in\mathcal{M}^{(2)}_{u,v}} \bar{\varphi}_{m,u,v} + \sum_{m\in\mathcal{M}^{(3)}_{u,v}} \bar{\psi}_{m,u,v}, \quad \omega_{v,v} \propto \sum_{m\in\mathcal{M}^{(2)}_v} \bar{\varphi}_{m,v,v}$$

where $\mathcal{M}^{(1)}_{u,v}$, $\mathcal{M}^{(2)}_{u,v}$ and $\mathcal{M}^{(3)}_{u,v}$ are defined by

$$\mathcal{M}^{(1)}_{u,v} = \{m \in \{1, \cdots, M\};\ v \in C_m,\ u \in \mathcal{B}_{m,v}\},$$
$$\mathcal{M}^{(2)}_{u,v} = \{m \in \{1, \cdots, M\};\ v \in \partial C_m,\ u \in B(v) \setminus \mathcal{B}_{m,v}\},$$
$$\mathcal{M}^{(3)}_{u,v} = \{m \in \{1, \cdots, M\};\ v \in \partial C_m,\ u \in \mathcal{B}_{m,v}\}.$$

The actual values are obtained after normalization. Recall that we can regard our estimation method as a kind of the EM algorithm. It should be noted that each time the iteration proceeds the value of the likelihood function never decreases and the iterative algorithm is guaranteed to converge.

3.3 Model Selection

Next, we describe our model selection method. We select the model based on predictive accuracy. Here, note that we cannot use an information theoretic criterion such as AIC (Akaike Information Criterion) or MDL (Minimum Description Length) because we need to select one from models with completely different probability distributions. Moreover, for both models, it is quite difficult to efficiently calculate the exact activation probability of each node even a few information diffusion cascading steps ahead. In oder to avoid these difficulties, we propose a method based on a hold-out strategy, which attempts to predict the activation probabilities at one step later.

For simplicity, we assume that for each D_m, the initial observation time t_m is zero, i.e., $t_m = 0$ for $m = 1, \cdots, M$. Then, we introduce a set of observation periods

$$\mathcal{I} = \{[0, \tau_n);\ n = 1, \cdots, N\},$$

where N is the number of observation data we want to predict sequentially and each τ_n has the following property: There exists some $(v, t_{m,v}) \in D_m$ such that $0 < \tau_n < t_{m,v}$. Let $D_{m;\tau_n}$ denote the observation data in the period $[0, \tau_n)$ for the mth diffusion result, i.e.,

$$D_{m;\tau_n} = \{(v, t_{m,v}) \in D_m;\ t_{m,v} < \tau_n\}.$$

We also set $\mathcal{D}_{M;\tau_n} = \{(D_{m;\tau_n}, \tau_n);\ m = 1, \cdots, M\}$. Let $\boldsymbol{\Theta}$ denote the set of parameters for either the AsIC or the AsLT models, i.e., $\boldsymbol{\Theta} = (\boldsymbol{r}, \boldsymbol{\kappa})$ or $\boldsymbol{\Theta} = (\boldsymbol{r}, \boldsymbol{\omega})$. We can estimate the values of $\boldsymbol{\Theta}$ from the observation data $\mathcal{D}_{M;\tau_n}$ by using the learning algorithms in Sections 3.1 and 3.2. Let $\widehat{\boldsymbol{\Theta}}_{\tau_n}$ denote the estimated values of $\boldsymbol{\Theta}$. Then, we can calculate the activation probability $q_{\tau_n}(v, t)$ of node v at time t ($\geq \tau_n$) using $\widehat{\boldsymbol{\Theta}}_{\tau_n}$.

For each τ_n, we select the node $v(\tau_n)$ and the time $t_{m(\tau_n),v(\tau_n)}$ by

$$t_{m(\tau_n),v(\tau_n)} = \min\left\{t_{m,v};\ (v, t_{m,v}) \in \bigcup_{m=1}^{M}(D_m \setminus D_{m;\tau_n})\right\}.$$

Note that $v(\tau_n)$ is the first active node in $t \geq \tau_n$. We evaluate the predictive performance for the node $v(\tau_n)$ at time $t_{m(\tau_n),v(\tau_n)}$. Approximating the empirical distribution by

$$p_{\tau_n}(v, t) \ = \ \delta_{v,v(\tau_n)}\, \delta(t - t_{m(\tau_n),v(\tau_n)})$$

with respect to $(v(\tau_n), t_{m(\tau_n),v(\tau_n)})$, we employ the Kullback-Leibler (KL) divergence

$$KL(p_{\tau_n} \,\|\, q_{\tau_n}) \ = \ -\sum_{v \in V} \int_{\tau_n}^{\infty} p_{\tau_n}(v, t) \log \frac{q_{\tau_n}(v, t)}{p_{\tau_n}(v, t)}\, dt,$$

where $\delta_{v,w}$ and $\delta(t)$ stand for Kronecker's delta and Dirac's delta function, respectively. Then, we can easily show

$$KL(p_{\tau_n} \,\|\, q_{\tau_n}) \ = \ -\log h_{m(\tau_n),v(\tau_n)}. \tag{8}$$

By averaging the above KL divergence with respect to $\mathcal{I}$, we propose the following model selection criterion $\mathcal{E}$ (see Equation (8)):

$$\mathcal{E}(\mathcal{X}; D_1 \cup \cdots \cup D_M) = -\frac{1}{N} \sum_{n=1}^{N} \log h_{m(\tau_n),v(\tau_n)}, \tag{9}$$

where $\mathcal{X}$ expresses the information diffusion model (i.e., the AsIC or the AsLT models). In our experiments, we adopted

$$\mathcal{I} = \{[0, t_{m,v}); (v, t_{m,v}) \in D_1 \cup \cdots \cup D_M, t_{m,v} \geq \tau_0\},$$

where τ_0 is the median time of all the observed activation time points.

3.4 Behavioral Analysis

Thus far, we assumed that $\boldsymbol{\Theta}$ can vary with respect to nodes and links but is independent of the topic of information diffused. However, they may be sensitive to the topic. We follow [13] and place a constraint that $\boldsymbol{\Theta}$ depends only on topics but not on nodes and links of the network G, and assign a different m to a different topic. Therefore, we set $r_{m,u,v} = r_m$ and $\kappa_{m,u,v} = \kappa_m$ for any link $(u, v) \in E$ in case of the AsIC model and $r_{m,v} = r_m$ and $\omega_{m,u,v} = q_m |B(v)|^{-1}$ for any node $v \in V$ and link $(u, v) \in E$ in case of the AsLT model. Here note that $0 < q_m < 1$ and $\omega_{v,v} = 1 - q_m$. Without this constraint, we only have one piece of observation for each (m, u, v) and there is no way to learn $\boldsymbol{\Theta}$.

Using each pair of the estimated parameters, (r_m, q_m) for the AsLT model and (r_m, κ_m) for the AsIC model, we can discuss which model is more appropriate for each topic, and analyze the behavior of people with respect to the topics of information by simply plotting them as a point in 2-dimensional space.

4 Performance Evaluation by Artificial Data

Our goal here is to evaluate the parameter learning and model selection methods to see how accurately it can detect the true model that generated the data, using topological structure of four large real networks. Here, we assumed the true model by which the data are generated to be either AsLT or AsIC.

4.1 Data Sets

We employed four datasets of large real networks (all bidirectionally connected). The first one is a trackback network of Japanese blogs used in [8] and has 12,047 nodes and 79,920 directed links (the blog network). The second one is a network of people derived from the "list of people" within Japanese Wikipedia, also used in [8], and has 9,481 nodes and 245,044 directed links (the Wikipedia network). The third one is a network derived from the Enron Email Dataset [16] by extracting the senders and the recipients and linking those that had bidirectional communications. It has 4,254 nodes and 44,314 directed links (the Enron network). The fourth one is a coauthorship network used in [17] and has 12,357 nodes and 38,896 directed links (the coauthorship network).

Here, according to [13], we assumed the simplest case where the parameter values are uniform across all links and nodes, i.e., $\omega_{u,v} = q|B(v)|^{-1}$, $r_v = r$ for AsLT, and $r_{u,v} = r$, $\kappa_{u,v} = \kappa$ for AsIC. Under this assumption there is no need for the observation sequence data to pass through every link or node at least once. This drastically reduces the amount of data necessary to learn the parameters. Then, our task is to estimate the

Table 1: Parameter estimation error of the learning method for four networks.

Network		Blog	Wiki	Enron	Coauthor
$\mathcal{D}_M(AsLT)$	r	0.248	0.253	0.200	0.244
	q	0.080	0.078	0.077	0.089
$\mathcal{D}_M(AsIC)$	r	0.114	0.026	0.029	0.167
	κ	0.020	0.013	0.002	0.054

Table 2: Accuracy of the model selection method for four networks.

Network	Blog	Wiki	Enron	Coauthor
$\mathcal{D}_M(AsLT)$	79	86	99	76
	(28.2)	(54.0)	(47.7)	(19.0)
$\mathcal{D}_M(AsIC)$	92	100	100	93
	(370.2)	(920.8)	(1500.6)	(383.5)

values of these parameters from data. The true value of q was set to 0.9 for every network to achieve reasonably long diffusion results, and the true value of r was set to 1.0. According to [7], we set κ to a value smaller than $1/\bar{d}$, where $\bar{d}$ is the mean out-degree of a network. Thus, the true value of κ was set to 0.2 for the coauthorship network, 0.1 for the blog and Enron networks, and 0.02 for the Wikipedia network. Using these values, two sets of data were generated for each network, one for the true AsLT model and the other for the true AsIC model, denoted by $\mathcal{D}_M(AsLT)$ and $\mathcal{D}_M(AsIC)$, respectively. For each of these, sequences of data were generated, each starting from a randomly selected initial active node and having at least 10 nodes. In our experiments, we set $M = 100$ and evaluated our model selection method in the framework of behavioral analysis. Parameter updating is terminated when either the iteration number reaches its maximum (set to 100) or the following condition is first satisfied: $|r(s+1) - r(s)| + |q(s+1) - q(s)| \leq 10^{-6}$ for AsLT, $|r(s+1) - r(s)| + |\kappa(s+1) - \kappa(s)| \leq 10^{-6}$ for AsIC. In most of the cases, the latter inequality is satisfied in less than 100 iterations. The converged values are rather insensitive to the initial values, and we confirmed that the parameter updating algorithm stably converges to the correct values. In actual computation, the learned values for τ_n is used as the initial values for τ_{n+1} for efficiency purpose.

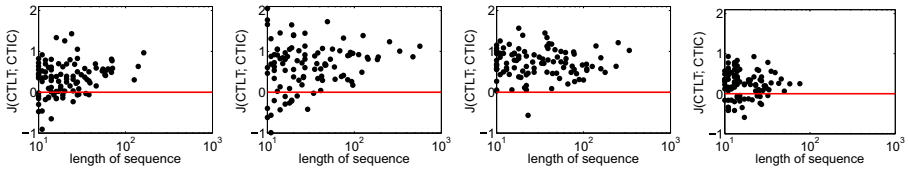
4.2 Learning Results

Table 1 shows the error in the estimated parameters for four networks by the proposed learning method. In this evaluation we treated each sequence as a separate observation and learned the parameters from each, repeated this M (=100) times and took the average. More specifically, the parameters of AsLT were estimated from $\mathcal{D}_M(AsLT)$, and those of AsIC from $\mathcal{D}_M(AsIC)$. Even though each pair of the parameters for individual models was estimated by using only one sequence data, we can see that the estimated values were reasonably close to the true one. This confirms that our proposed learning methods work well. The results indicate that the estimation performance on AsIC is substantially better than that on AsLT. We consider that this performance difference is attributed to the average sequence length, as discussed later.

4.3 Model Selection Results

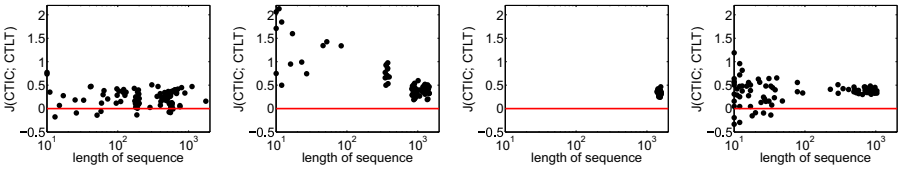
The average KL divergence given by equation (9) is the measure for the goodness of the model $\mathcal{X}$, given the data D_m. The smaller its value is, the better the model explains the data in terms of predictability. Thus, we can estimate the true model from which D_m is generated to be AsLT if $\mathcal{E}(AsLT; D_m) < \mathcal{E}(AsIC; D_m)$, and vice versa.

Table 2 summarizes the number of sequences for which the model selection method correctly identified the true model. The number within the parentheses is the average



(a) Blog (b) Wikipedia (c) Enron (d) Coauthorship

Fig. 1: Relation between the length of sequence and the the accuracy of model selection for $\mathcal{D}_M(AsLT)$.



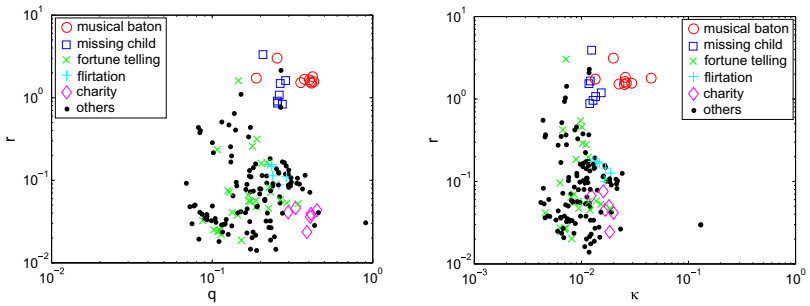
(a) Blog (b) Wikipedia (c) Enron (d) Coauthorship

Fig. 2: Relation between the length of sequence and the the accuracy of model selection for $\mathcal{D}_M(AsIC)$.

length of the sequences in each dataset. From these results, we can say that the proposed method achieved a good accuracy, 90.6% on average. Especially, for the Enron network, its estimation was almost perfect. To analyze the performance of the proposed method more deeply, we investigated the relation between the length of sequence and the model selection result. It is shown in Fig. 1 for $\mathcal{D}_M(AsLT)$, where the horizontal axis denotes the length of sequence in each dataset and the vertical axis is the difference of the average KL divergence defined by $J(AsLT;AsIC) = \mathcal{E}(AsIC;D_m) - \mathcal{E}(AsLT;D_m)$. Thus, $J(AsLT;AsIC) > 0$ means that the proposed method correctly estimated the true model for the dataset $D_m(AsLT)$ because it means $\mathcal{E}(AsLT;D_m)$ is smaller than $\mathcal{E}(AsIC;D_m)$. From these figures, we can see that there is a correlation between the length of sequence and the estimation accuracy, and that the misselection occurs only in short sequences for every network. We notice that the overall accuracy becomes 95.5% when considering only the sequences that contain no less than 20 nodes. This means that the proposed model selection method is highly reliable for a long sequence and its accuracy could asymptotically approach to 100% as the sequence gets longer. Figure 2 is the results for $\mathcal{D}_M(AsIC)$, where $J(AsIC;AsLT) = \mathcal{E}(AsLT;D_m) - \mathcal{E}(AsIC;D_m)$. The results are better than for $\mathcal{D}_M(AsLT)$. In particular, Wikipedia and Blog networks have no misselection. We note that the plots are shifted to the right for all networks, meaning that the data sequences are longer for $\mathcal{D}_M(AsIC)$ than for $\mathcal{D}_M(AsLT)$. The better accuracy is attributed to this.

5 Behavioral Analysis of Real World Blog Data

We analyzed the behavior of topics in a real world blog data. Here, again, we assumed the true model behind the data to be either AsLT or AsIC. Then, we first applied our



(a) the AsLT model based method (b) the AsIC model based method

Fig. 3: Results for the Doblog database.

learning method to behavioral analysis based on the method described in Section 3.4, assuming two possibilities, i.e. the true model being either AsLT or AsIC for all the topics, and investigated how each topic spreads throughout the network by comparing the learned parameter values. Next, we estimated the true model of each data sequence for each topic by applying the model selection method described in Section 3.3.

5.1 Data Sets

We used the real blogroll network used in [13], which was generated from the database of a blog-hosting service in Japan called *Doblog* [1]. In the network, bloggers are connected to each other and we assume that topics propagate from blogger x to another blogger y when there is a blogroll link from y to x. In addition, according to [18], it is assumed that a topic is represented as a URL which can be tracked down from blog to blog. We used the propagation sequences of 172 URLs for this analysis, each of which has at least 10 time steps. Please refer to [13] for more details.

5.2 Behavioral Analysis

We ran the experiments for each identified URL and obtained the parameters q and r for the AsLT model based method and κ and r for the AsIC model based method. Figures 3a and 3b are the plots of the results for the major URLs (topics) by the AsLT and AsIC methods, respectively. The horizontal axis is the diffusion parameter q for the AsLT method and κ for the AsIC method, while the vertical axis is the delay parameter r for both. The latter axis is normalized such that $r = 1$ corresponds to a delay of one day, meaning $r = 0.1$ corresponds to a delay of 10 days. In these figures, we used five kinds of markers other than dots, to represent five different typical URLs: the circle ($\circ$) stands for a URL that corresponds to the musical baton which is a kind of telephone game on the Internet (the musical baton), the square ($\square$) for a URL that corresponds to articles about a missing child (the missing child), the cross ($\times$) for a URL that corresponds to articles about fortune telling (the fortune telling), the diamond ($\diamond$) for a URL of a certain charity site (the charity), and the plus ($+$) for a URL of a site for flirtatious

[1] Doblog(`http://www.doblog.com/`), provided by NTT Data Corp. and Hotto Link, Inc.

tendency test (the flirtation). All the other topics are denoted by dots (·), which means they are a mixture of many topics.

The results indicates that in general both the AsLT and AsIC models capture reasonably well the characteristic properties of topics in a similar way. For example, it captures the urgency of the missing child, which propagates quickly. Musical baton which actually became the latest craze on the Internet also propagates quickly. In contrast non-emergency topics such as the flirtation and the charity propagate very slowly. Unfortunately, this highlights the people's low interest level of the charity activity in the real world. We further note that the dependency of topics on the parameter r is almost the same for both AsLT and AsIC, but that on the parameters q and κ is slightly different, e.g., relative difference of musical baton, missing child and charity. Although q and κ are different parameters but both are the measures that represent how easily the diffusion takes place. We showed in [13] that the influential nodes are very sensitive to the model used and this can be attributed to the differences of these parameter values.

5.3 Model Selection

In the analysis of previous subsection, we assumed that each topic follows the same diffusion model. However, in reality this is not true and each topic should propagate following more closely to either one of the AsLT and AsIC models. Thus, in this subsection, we attempt to estimate the underlying behavior model of each topic by applying the model selection method to individual sequence as described in section 4. Namely, we regard that each observation consists of only one observed data sequence, i.e., $\mathcal{D}_1$, and calculate its KL divergences by equation (9) for the both models, and compare the goodness.

Table 3 and Fig. 4 summarize the results. From these results, we can see that most of the diffusion behaviors on this blog network follows the AsIC model. It is interesting to note that the model estimated for the musical baton is not identical to that for the missing child although their diffusion patterns are very similar in the previous analysis. The missing child strictly follows the AsIC model. This is attributed to its greater urgency. On the other, musical baton seems to follow more closely to AsLT. This is because the longer sequence results in a better accuracy and the models selected in longer sequences are all AsLT in Fig. 4 although the numbers are almost tie (4 vs. 5) in Table 3. This can be interpreted that people follow their friends in this game. Likewise, it is easy to imagine that one would align oneself with the opinions of those around when requested to raise funds. This explains that charity follows AsLT. The flirtation clearly follows AsLT. This is probably because the information of this kind of play site easily diffuses within close friends. Note that there exists one dot at near the top center in Fig. 4, showing the greatest tendency to follow AsLT. This dot represents a typical circle site that distributes one's original news article on personal events.

6 Discussion

We now have ways to compare the diffusion process with respect to two models (the AsLT model and the AsIC model) for the same observed dataset. Being able to learn the

Table 3: Results of model selection for the Doblog dataset.

Topic	Total	AsLT	AsIC
Musical baton	9	5	4
Missing child	7	0	7
Fortune telling	28	4	24
Charity	6	5	1
Flirtation	7	7	0
Others	115	11	104

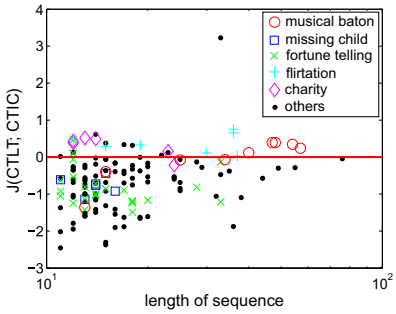


Fig. 4: The relation between the KL difference and sequence length for the Doblog database.

parameters of these models enable us to analyze the diffusion process more precisely. Comparing the results bring us deeper insights into the relation between models and information diffusion processes.

We note that the formulation in Sections 2 and 3 allows the parameters to depend on links and nodes, but the analysis we showed in Section 4 is for the simplest case where the parameters are uniform across the whole network. Actually, if all the parameters are node and link dependent, the number of the parameters becomes so huge and it is not practical (almost impossible) to estimate them accurately because the amount of observation data needed is prohibitively huge and there is always a problem of overfitting. However, this can be alleviated. In a more realistic setting we can divide E into subsets $E_1, E_2, ..., E_L$ and assign the same value for each parameter within each subset. For example, we may divide the nodes into two groups: those that strongly influence others and those not, or we may divide the nodes into another two groups: those that are easily influenced by others and those not. If there is some background knowledge about the node grouping, our method can make the best use of it. Obtaining such background knowledge is also an important research topic in the knowledge discovery from social networks.

The discussion above is also related to the use of the data for model selection in Section 5 in which we used each sequence separately to learn the model parameter values and select the model rather than using them altogether for the same topic and obtaining a single set of parameter values. The results in Section 5 show that the model parameters thus obtained for each sequence are very similar to each other for the same topic. This in turn justifies the use of the same parameter values for multiple sequence observation data (the way we formulated in Section 3.3).

As we mentioned in Section 5.2 but did not show in this paper due to the space limitation, the ranking results that involve detailed probabilistic simulation is very sensitive to the underlying model which is assumed to generate the observed data. In other words, it is very important to select an appropriate model for the analysis of information diffusion from which the data has been generated if the node characteristics are the main objective of analysis, e.g. such problems as the influence maximization problem [7, 11], a problem at a more detailed level. However, it is also true that the parameters for

the topics that actually propagated quickly/slowly in observation converged to the values that enable them to propagate quickly/slowly on the model, regardless of the model chosen. Namely, we can say that the difference of models does not have much influence on the relative difference of topic propagation which indeed strongly depends on topic itself. Both models are well defined and can explain this property at this level of abstraction. Nevertheless, the model selection is very important if we want to characterize how each topic propagates through the network.

Finally, the proposed learning method is efficient and the runtime is not an issue. The convergence is fast and it can handle networks of millions of nodes.

7 Conclusion

We considered the problem of analyzing information diffusion process in a social network using two kinds of information diffusion models, incorporating asynchronous time delay, the AsLT model and the AsIC model, and investigated how the results differ according to the model used. To this end, we proposed novel methods of 1) learning the parameters of the AsLT model from the observed data (the method for learning the parameters of the AsIC model has already been reported), and 2) selecting models that better explains the observation. We experimentally confirmed that the learning method converges to the correct values very stably and the model selection method can correctly identifies the diffusion models by which the observed data is generated based on extensive simulations on four real world datasets. We further applied the methods to the real blog data and analyzed the behavior of topic propagation. The relative propagation speed of topics, i.e. how far/near and how fast/slow each topic propagates, that are derived from the learned parameter values is rather insensitive to the model selected, but the model selection algorithm clearly identifies the difference of model goodness for each topic. We found that many of the topics follow the AsIC models in general, but some specific topics have clear interpretations for them being better modeled by either one of the two, and these interpretations are consistent with the model selection results. There are numerous factors that affects the information diffusion process, and there can be a number of different models. Model selection is a big challenge in social network analysis and this work is the first step towards this goal.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Physical Review E **66** (2002) 035101
2. Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45** (2003) 167–256

3. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. SIGKDD Explorations **6** (2004) 43–52
4. Domingos, P.: Mining social networks for viral marketing. IEEE Intelligent Systems **20** (2005) 80–82
5. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06). (2006) 228–237
6. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters **12** (2001) 211–223
7. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003). (2003) 137–146
8. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data **3** (2009) 9:1–9:23
9. Watts, D.J.: A simple model of global cascades on random networks. Proceedings of National Academy of Science, USA **99** (2002) 5766–5771
10. Watts, D.J., Dodds, P.S.: Influence, networks, and public opinion formation. Journal of Consumer Research **34** (2007) 441–458
11. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07). (2007) 1371–1376
12. Saito, K., Kimura, M., Nakano, R., Motoda, H.: Finding influential nodes in a social network from information diffusion data. In: Proceedings of the International Workshop on Social Computing and Behavioral Modeling (SBP09). (2009) 138–145
13. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Learning continuous-time information diffusion model for social behavioral data analysis. In: Proceedings of the 1st Asian Conference on Machine Learning (ACML2009). (2009) 322–337
14. Goyal, A., Bonchi, F., Lakshhmanan, L.V.S.: Learning influence probabilities in social networks. In: Proceedings of the third ACM international conference on Web Search and Data Mining. (2010) 241–250
15. Bakshy, E., Karrer, B., Adamic, L.A.: Social influence and the diffusion of user-created content. In: Proceedings of the tenth ACM conference on Electronic Commerce. (2009) 325–334
16. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: Proceedings of the 2004 European Conference on Machine Learning (ECML'04). (2004) 217–226
17. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435** (2005) 814–818
18. Adar, E., Adamic, L.A.: Tracking information epidemics in blogspace. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05). (2005) 207–214

Acquiring Expected Influence Curve from Single Diffusion Sequence

Yuya Yoshikawa[1], Kazumi Saito[1], Hiroshi Motoda[2], Masahiro Kimura[3], and Kouzou Ohara[4]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
{b7101,k-saito}@u-shizuoka-ken.ac.jp
[2] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp
[3] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[4] Department of Integrated Information Technology, Aoyama Gakuin Univesity
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp

Abstract. We address the problem of estimating the expected influence curves with good accuracy from a single observed information diffusion sequence, for both the asynchronous independent cascade (AsIC) model and the asynchronous linear threshold (AsLT) model. We solve this problem by first learning the model parameters and then estimating the influence curve using the learned model. Since the length of the observed diffusion sequence may vary from a very long one to a very short one, we evaluate the proposed method by simulation using artificial diffusion sequence of various lengths and show that the proposed method can estimate the expected influence curve robustly from a single diffusion sequence with various lengths.

1 Introduction

The rise of the Internet and the World Wide Web accelerates the creation of various large-scale social networks, and considerable attention has been brought to social networks as an important medium for the spread of information [1–5]. Innovation, topics and even malicious rumors can diffuse through social networks in the form of so-called "word-of-mouth" communications. Such social interaction processes are usually characterized by highly distributed phenomena over a social network, but the complexity and distributed nature of these processes does not necessarily imply that these evolutions are chaotic or unpredictable. Just as natural scientists discover laws and create models for their fields, so can one, in principle, find empirical regularities and develop explanatory accounts of evolution in a social network. Especially, such predictive knowledge would be valuable for market opportunities. In this paper, as a piece of such predictive knowledge, we focus on acquiring the expected influence curve of each information source node by using information diffusion models.

Widely used information diffusion models in recent studies are the *independent cascade (IC)* [6–8] and the *linear threshold (LT)* [9, 10]. They have been used to solve such problems as the *influence maximization problem* [7, 11]. These two models focus on different information diffusion aspects. The IC model is sender-centered and an active node influences its inactive neighbors *independently* with diffusion probabilities assigned to links. On the other hand, the LT model is receiver-centered and a node is influenced by its active neighbors if the sum of their weights exceeds the threshold for the node. Both models have parameters that need be specified in advance: diffusion probabilities for the IC model, and weights for the LT model. However, their true values are not known in practice. This poses yet another problem of estimating them from a set of information diffusion results that are observed as time-sequences of influenced (activated) nodes. To the best of our knowledge, there are only a few methods that can estimate the parameter values for the IC and LT models and their variants that incorporate asynchrnous time delay (referred to as the AsIC model and the AsLT model) [3, 12–14]. We follow the methods in [13, 14] in this paper.

Now assume that we observed a single information diffusion sequence for an information source node. How can we acquire the expected influence curve from this single instance of observation? This is the problem we want to solve. In a sence, this sequence can be regarded as a piece of crude knowledge about the expected influence curve because we can count the number of nodes that have been influenced (activated) by any time point t which we specify. However, due to its stochastic nature, such a sequence varies in a quite wide range each time we observe it, even if we know which of the two models (AsIC and AsLT) the information diffusion follows. Thus, it is undesirable to approximate the expected influence curve by a single instance of observed sequence.

In this paper, we assume that information diffuses over a network by either the AsIC model or the AsLT model, and propose a novel method for estimating the expected influence curve by first estimating parameters for the assumed models from a single observed informaition diffusion sequence and use the learned model to estimate the expected curve. In another word, our method can be viewed as a knowledge refinement method from the observed single information diffusion sequence to the expected influence curve based on the information diffusion model. We performed extensive experiments to evaluate whether the proposed method can estimate the influence curve much more accurately than the observed diffusion curve itself. The results cleary show the advantage of our method.

The paper is organized as follows. We revisit the information diffusion models and briefly explain the independent cascade model, the linear threshold model, and their asynchronous time delay versions (the models we use in this paper) : AsCT and AsLT in section 2, and revisit parameter learning algorithms for AsCT and AsLT in section 3. We then describe the estimation method of the expected influence curve in section 4, and explain the experimental results in detail in section 5, followed by some discussions in section 6. We summarize our conlcusion in section 7.

2 Information Diffusion Models

We first define the IC model according to [7], and then introduce the asynchronous IC model (AsIC). After that, we do the same for the LT model and the asynchronous LT model (AsLT). We mathematically model the spread of information over a directed network $G = (V, E)$ without self-links, where V and E ($\subset V \times V$) stands for the sets of all the nodes and links, respectively. We call nodes *active* if they have been influenced with the information. It is assumed that nodes can switch their states only from inactive to active, but not from active to inactive. Given an initial set S of active nodes, we assume that the nodes in S have first become active at an initial time, and all the other nodes are inactive at that time. Node u is called a *child node* of node v if $(v, u) \in E$, and node u is called a *parent node* of node v if $(u, v) \in E$. For each node $v \in V$, let $F(v)$ and $B(v)$ denote the set of child nodes of v and the set of parent nodes of v, respectively,

$$F(v) = \{w \in V;\ (v, w) \in E\}, \quad B(v) = \{u \in V;\ (u, v) \in E\}.$$

2.1 Independent Cascade Model

The IC model is a fundamental probabilistic model for the spread of a disease. In this model, we specify a real value $\kappa_{u,v}$ with $0 < \kappa_{u,v} < 1$ for each link (u, v) in advance. Here $\kappa_{u,v}$ is referred to as the *diffusion probability* through link (u, v). The diffusion process unfolds in discrete time-steps $t \geq 0$, and proceeds from a given information source node in the following way. When a node u becomes active at time-step t, it is given a single chance to activate each currently inactive child node v, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time-step $t + 1$. If multiple parent nodes of v become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

2.2 Asynchronous Independent Cascade Model

Next, we extend the IC model so as to allow continuous-time delays, and refer to the extended model as the *Asynchronous independent cascade (AsIC) model*. In the AsIC model, we specify a real value $r_{u,v}$ with $r_{u,v} > 0$ for each link $(u, v) \in E$ in advance together with diffusion parameter $\kappa_{u,v}$. We refer to $r_{u,v}$ as the *time-delay parameter* through link (u, v).

The diffusion process unfolds in continuous-time t, and proceeds from a given information source node in the following way. Suppose that a node u becomes active at time t. Then, node u is given a single chance to activate each currently inactive child node v. We choose a delay-time δ from the exponential distribution with parameter $r_{u,v}$. If node v is not active before time $t + \delta$, then node u attempts to activate node v, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time $t + \delta$. Under the continuous time framework, it is unlikely that multiple parent nodes of v attempt to activate v at exactly the same time $t + \delta$. So we ignore this possibility. Whether or not

u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

For an information source node v, let $\varphi(t; v)$ denote the number of active nodes at a specified time t, i.e. the number of nodes that have become activated by t. Note that $\varphi(t; v)$ is a random variable. Let $\sigma(t; v)$ denote the expected value of $\varphi(t; v)$. We call $\sigma(t; v)$ the *expected influence curve* of v for the AsIC model.

2.3 Linear Threshold Model

The LT model is a fundamental probabilistic model for the spread of innovation. In this model we specify a *weight* ($\omega_{u,v} > 0$) for every node $v \in V$ from its parent node u in advance such that $\sum_{u \in B(v)} \omega_{u,v} \leq 1$. The diffusion process from a given initial active set S proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0, 1]$. At time-step t, an inactive node v is influenced by each of its active parent nodes, u, according to weight $\omega_{u,v}$. If the total weight from active parent nodes of v is no less than threshold θ_v, that is, $\sum_{u \in B_t(v)} \omega_{u,v} \geq \theta_v$, then v will become active at time-step $t + 1$. Here, $B_t(v)$ stands for the set of all the parent nodes of v that are active at time-step t. The process terminates if no more activations are possible.

2.4 Asynchronous Linear Threshold Model

We make a similar extention to the LT model so as to allow continuous-time delays, and refer to the extended model as the *Asynchronous linear threshold (AsLT) model*. In the AsLT model, in addition to the weight set $\{\omega_{u,v}\}$, we specify real values r_v with $r_v > 0$ in advance for each node $v \in V$. We refer to r_v as the *time-delay parameter* on node v. Note that r_v depends only on v, which means that it is the node v's decision when to receive the information once the activation condition has been satisfied.

The diffusion process unfolds in continuous-time t, and proceeds from a given initial active set S in the following way. Suppose that the total weight from active parent nodes of v became no less than the threshold θ_v at time t for the first time. Then, v will become active at time $t + \delta$, where we choose a delay-time δ from the exponential distribution with parameter r_v. Further, note that even though some other non-active parent nodes of v become active during the time period between t and $t + \delta$, the activation time of v, $t + \delta$, still remains the same. The other diffusion mechanisms are the same as the LT model. Similarly to the AsIC model, we can also define the expected influence curve $\sigma(t; v)$ of an information source node v for the AsIC model.

3 Learning Algorithms

We define the time-delay parameter vector $\boldsymbol{r}$ and the diffusion parameter vector $\boldsymbol{\kappa}$ by $\boldsymbol{r} = (r_{u,v})_{(u,v)\in E}$ and $\boldsymbol{\kappa} = (\kappa_{u,v})_{(u,v)\in E}$ for the AsIC model,. Similarly, we define the parameter vectors $\boldsymbol{\omega}$ and $\boldsymbol{r}$ by $\boldsymbol{\omega} = (\omega_{u,v})_{(u,v)\in E}$ and $\boldsymbol{r} = (r_v)_{v\in V}$ for the AsLT model. In practice, the true values of these parameters are not available. Thus, we must learn them from past information diffusion histories.

We consider an observed data set of M independent information diffusion results, $\{D_m;\ m = 1, \cdots, M\}$. Here, each D_m is a set of pairs of active nodes and their activation times in the mth information diffusion result, $D_m = \{(u, t_{m,u}),\ (v, t_{m,v}),\ \cdots\}$. For each D_m, we denote the observed initial time by $t_m = \min\{t_{m,v};\ (v, t_{m,v}) \in D_m\}$, and the observed final time by $T_m \geq \max\{t_{m,v};\ (v, t_{m,v}) \in D_m\}$. Note that T_m is not necessarily equal to the final activation time. Hereafter, we express our observation data by $\mathcal{D}_M = \{(D_m, T_m);\ m = 1, \cdots, M\}$. For any $t \in [t_m, T_m]$, we set $C_m(t) = \{v;\ (v, t_{m,v}) \in D_m,\ t_{m,v} < t\}$. Namely, $C_m(t)$ is the set of active nodes before time t in the mth information diffusion result. For convenience sake, we use C_m as referring to the set of all the active nodes in the mth information diffusion result. Moreover, we define a set of non-active nodes with at least one active parent node for each by $\partial C_m = \{v;\ (u, v) \in E,\ u \in C_m,\ v \notin C_m\}$. For each node $v \in C_m \cup \partial C_m$, we define the following subset of parent nodes, each of which has a chance to activate v.

$$\mathcal{B}_{m,v} = \begin{cases} B(v) \cap C_m(t_{m,v}) & \text{if } v \in C_m(t_{m,v}), \\ B(v) \cap C_m & \text{if } v \in \partial C_m. \end{cases}$$

In oder to learn the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ for the AsIC model, and the values of $\boldsymbol{r}$ and $\boldsymbol{\omega}$ for the AsLT model for the given $\mathcal{D}_M$, we adopt the method proposed in [13] and [14], respectively, each of which is only briefly explained here.

3.1 Learning Parameters of AsIC Model

To learn the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$ for the AsIC model, we revisit the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\omega}$ to use as the objective function [13]. First, we consider any node $v \in C_m$ with $t_{m,v} > t_m$ for the mth information diffusion result. Let $\Phi_{m,u,v}$ denote the probability density that a node $u \in B(v) \cap C_m(t_{m,v})$ activates the node v at time $t_{m,v}$, that is,

$$\Phi_{m,u,v} = \kappa_{u,v} r_{u,v} \exp(-r_{u,v}(t_{m,v} - t_{m,u})). \tag{1}$$

Let $\Psi_{m,u,v}$ denote the probability that the node v is not activated from a node $u \in B(v) \cap C_m(t_{m,v})$ during the time-period $[t_{m,u}, t_{m,v}]$, that is,

$$\begin{aligned} \Psi_{m,u,v} &= 1 - \kappa_{u,v} \int_{t_{m,u}}^{t_{m,v}} r_{u,v} \exp(-r_{u,v}(t - t_{m,u}))dt \\ &= \kappa_{u,v} \exp(-r_{u,v}(t_{m,v} - t_{m,u})) + (1 - \kappa_{u,v}). \end{aligned} \tag{2}$$

As explained in 2.2, it is not necessary to consider simultaneous activations by multiple active parents even if $\eta = |B(v) \cap C_m(t_{m,v})| > 1$. Thus, the probability density that the node v is activated at time $t_{m,v}$, denoted by $h^{(IC)}_{m,v}$, can be expressed as

$$\begin{aligned} h^{(IC)}_{m,v} &= \sum_{u \in B(v) \cap C_m(t_{m,v})} \Phi_{m,u,v} \left(\prod_{x \in B(v) \cap C_m(t_{m,v}) \setminus \{u\}} \Psi_{m,x,v} \right) \\ &= \prod_{x \in B(v) \cap C_m(t_{m,v})} \Psi_{m,x,v} \sum_{u \in B(v) \cap C_m(t_{m,v})} \Phi_{m,u,v} (\Psi_{m,u,v})^{-1}. \end{aligned} \tag{3}$$

Note that we are not able to know which node u actually activated the node v. This can be regarded as a hidden structure.

Next, for the mth information diffusion result, we consider any link $(v, w) \in E$ such that $v \in C_m$ and $w \notin C_m$. Let $g^{(IC)}_{m,v,w}$ denote the probability that the node w is not activated by the node v during the observed time period $[t_m, T_m]$. We can easily derive the following equation:

$$g^{(IC)}_{m,v,w} = \kappa_{v,w} \exp(-r_{v,w}(T_m - t_{m,v})) + (1 - \kappa_{v,w}). \tag{4}$$

Therefore, by using equations (3), (4), and the independence properties, we can define the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ by

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M) = \prod_{m=1}^{M} \prod_{v \in C_m} \left(h^{(IC)}_{m,v} \prod_{w \in F(v) \setminus C_m} g^{(IC)}_{m,v,w} \right), \tag{5}$$

Thus, our problem is to obtain the time-delay parameter vector $\boldsymbol{r}$ and the diffusion parameter vector $\boldsymbol{\kappa}$, which together maximize Equation (5). To obtain the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, we can employ a learning method based on the Expectation-Maximization algorithm in order to stably obtain its solutions [13].

3.2 Learning Parameters of AsLT Model

To learn the values of $\boldsymbol{r}$ and $\boldsymbol{\omega}$ from $\mathcal{D}_M$ for the AsLT model, we also revisit the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\omega}$ to use as the objective function [14]. For the sake of technical convenience, we introduce a slack weight $\omega_{v,v}$ for each node $v \in V$ such that $\omega_{v,v} + \sum_{u \in B(v)} \omega_{u,v} = 1$. Here note that such a slack weight $\omega_{v,v}$ never contributes to the activation of v and that for each node v, since a threshold θ_v is chosen uniformly at random from the interval $[0, 1]$, we can regard each weight $\omega_{*,v}$ as a multinomial probability.

Suppose that a node v became active at time $t_{m,v}$ for the mth result. Then, we know that the total weight from active parent nodes of v became no less than the threshold θ_v at the time when one of these active parent nodes, $u \in \mathcal{B}_{m,v}$, became first active. However, in case of $|\mathcal{B}_{m,v}| > 1$, there is no way of exactly knowing the actual nodes due to the continuous time-delay. Suppose that a node v was actually activated when a node $\zeta \in \mathcal{B}_{m,v}$ became activated. Then θ_v is between $\sum_{u \in B(v) \cap C_m(t_{m,\zeta})} \omega_{u,v}$ and $\omega_{\zeta,v} + \sum_{u \in B(v) \cap C_m(t_{m,\zeta})} \omega_{u,v}$. Namely, the probability that θ_v is chosen from this range is $\omega_{\zeta,v}$. Here note that such events with respect to different active parent nodes are mutually disjoint. Thus, the probability density that the node v is activated at time $t_{m,v}$, denoted by $h^{(LT)}_{m,v}$, can be expressed as

$$h^{(LT)}_{m,v} = \sum_{u \in \mathcal{B}_{m,v}} \omega_{u,v} r_v \exp(-r_v(t_{m,v} - t_{m,u})). \tag{6}$$

Here we define $h^{(LT)}_{m,v} = 1$ if $t_{m,v} = t_m$.

Next, we consider any node $w \in V$ belonging to $\partial C_m = \{w; (v, w) \in E \wedge v \in C_m(T_m) \wedge w \notin C_m(T_m)\}$ for the mth result. Let $g_{m,v}$ denote the probability that the node

v is not activated during the observed time period $[t_m, T_m]$. We can calculate $g_{m,v}$ as follows:

$$\begin{aligned} g_{m,v}^{(LT)} &= 1 - \sum_{u \in \mathcal{B}_{m,v}} \omega_{u,v} \int_{t_{m,u}}^{T_m} r_v \exp(-r_v(t - t_{m,u}))dt \\ &= 1 - \sum_{u \in \mathcal{B}_{m,v}} \omega_{u,v}(1 - \exp(-r_v(T_m - t_{m,v}))) \\ &= \omega_{v,v} + \sum_{u \in B(v) \setminus \mathcal{B}_{m,v}} \omega_{u,v} + \sum_{u \in \mathcal{B}_{m,v}} \omega_{u,v} \exp(-r_v(T_m - t_{m,v})). \end{aligned} \tag{7}$$

Therefore, by using Equations (6) and (7), and the independence properties, we can define the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\omega}$ by

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M) = \prod_{m=1}^{M} \left(\prod_{v \in C_m} h_{m,v}^{(LT)} \right) \left(\prod_{v \in \partial C_m} g_{m,v}^{(LT)} \right). \tag{8}$$

Thus, our problem is to obtain the time-delay parameter vector $\boldsymbol{r}$ and the diffusion parameter vector $\boldsymbol{\omega}$, which together maximize Equation (8). To obtain the values of $\boldsymbol{r}$ and $\boldsymbol{\omega}$, we can also employ a learning method based on the Expectation-Maximization algorithm in order to stably obtain its solutions [14].

4 Expected Influence Curve Acquisition

Thus far, we assumed that the time-delay and diffusion parameters can vary with respect to nodes and links. However, as mentioned earlier, we address the problem of estimating the influence curves from single observed diffusion sequences. Thus, in order to avoid overfitting to the observed data, we place a constraint that the parameters are uniform on nodes and links throughout the network G. Therefore, we set $r_{u,v} = r$ and $\kappa_{u,v} = \kappa$ for any link $(u, v) \in E$ in case of the AsIC model and $r_v = r$ and $\omega_{u,v} = \kappa |B(v)|^{-1}$ for any node $v \in V$ and link $(u, v) \in E$ in case of the AsLT model, where note that $0 < \kappa < 1$ and $\omega_{v,v} = 1 - \kappa$. Namely, since parameter κ of the AsLT model can be interpreted as a kind of diffusion probability, we employ the same symbol as used in the AsIC model. Without this constraint there is no way to learn the parameters since we only have one sequence of observation that covers only a small part of existing links.

We describe our method for acquiring an expected influence curve under the AsIC and AsLT model. Assume that we have observed the following single information diffusion sequence from the information source node v_0 at time t_0.

$$d = \{(v_0, t_0),\ (v_1, t_1),\ \cdots,\ (v_T, t_T)\}$$

First, by using the method described in Section 3.1 or 3.2, we can learn a pair of model parameters, κ and r, from the observed diffusion sequence d. Next, by using the method described in Section 2.2 or 2.4, we obtain the following K sets of simulated diffusion sequences

$$s_k = \{(v_0, t_0),\ (v_{k,1}, t_{k,1}),\ \cdots,\ (v_{k,T}, t_{k,T})\},\ k = 1,\ \cdots,\ K.$$

Here note that the information source node v_0 at time t_0 is the same for all sequences, but their final activation times $\{t_{k,T}\}$ as well as their numbers of activated nodes $\{|s_k|\}$ vary in quite wide range, as shown later in our experiments. Finally, by using the generated sequences $S = \{s_1, \cdots, s_K\}$, we can estimate the expected influence curve $\sigma(t, v_0)$ as follows:

$$\sigma(t; v_0, d) = \frac{1}{K} \sum_{k=1}^{K} |\{(v, \tau) \in s_k \, ; \tau \leq t\}| \tag{9}$$

This method needs three kinds of input information, i.e., the single observed diffusion sequence d, the topology of observed social network G, and the number of diffusion simulation trials K; then it outputs the expected influence curve $\sigma(t, v_0)$. Below we summarize the estimation algorithm.

step 1 Learn a pair of parameters κ and r from d.
step 2 Generate $S = \{s_1, ..., s_K\}$ by simulating information diffusion K times with the learned parameters κ and r.
step 3 Calculate the expected influence curve $\sigma(t; v_0, d)$ as the average of S.

In our experiments, the number of diffusion simulation trials is set to $K = 100$.

5 Experiments

We evaluate the feasibility of the proposed estimation method using the topologies of two large real network data.

5.1 Evaluation Procedure

Below we describe a procedure to evaluate our proposed method.

proc. 1 Decide information diffusion model: AsIC or AsLT, and choose its true parameters κ^* and r^*, and an information source node v_0 at time t_0.
proc. 2 Generate N sets of the diffusion sequences D under the setting of proc. 1.
proc. 3 Calculate the expected influence curve $\sigma(t; v_0)$ from D (by Equation (9) with S replaced by D) and the empirical influence curve $\varphi(t; v_0, d_n)$ from each $d_n \in D$.
proc. 4 Estimate the expected influence curve $\sigma(t; v_0, d_n)$ from each $d_n \in D$ by the proposed method in Section 4.
proc. 5 Calculate the RMSE curves E_C and E_D for evaluation.

In reality it is almost impossible to obtain the actual expected influence curve from observation. Thus our evaluation resorts to experiments based on synthetic data by assuming an information diffusion model, AsIC or AsLT, with a pair of model parameters, κ^* and r^* which we assume to be true (proc. 1). Then, by performing simulation based on the model with the true parameters, we can prepare N sets of synthetic diffusion sequences denoted by $D = \{d_1, \cdots, d_N\}$ (proc. 2). Next, by applying Equation (9) with respect to D (instead of S), we can obtain a reasonably accurate expected influence curve $\sigma(t; v_0)$ (proc. 3). Here we can also obtain an empirical influence curve for each of the generated sequence d_n defined by $\varphi(t; v_0, d_n) = |\{(v, \tau) \in d_n \, ; \tau \leq t\}|$ (proc. 3)[2]. On the other hand, by regarding each of the generated sequence d_n as a single ob-

[2] Note that d_n is not continuous but $\varphi(t; v_0, d_n)$ is continuous with respect to t.

served diffusion sequence, we can estimate the expected influence curve $\sigma(t; v_0, d_n)$ by our method proposed in Section 4 (proc. 4). Finally, we evaluate the average accuracy of the expected influence curves estimated by our method by means of the RMSE (Root Mean Squered Error) curve $E_C(t)$ and compare it with that of the empirical influence curves denoted by $E_D(t)$. Here these RMSE curves, $E_C(t)$ and $E_D(t)$, are defined as follows.

$$E_C(t) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\sigma(t; v_0, d_n) - \sigma(t; v_0))^2}, \quad E_D(t) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\varphi(t; v_0, d_n) - \sigma(t; v_0))^2}.$$

We can consider that the RMSE curve for $E_D(t)$ corresponds to the average accuracy of the single observed diffusion sequence when we interprete it as a piece of crude knowledge.

5.2 Experimental Settings

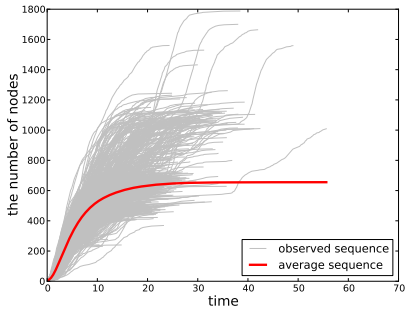
In our experiments, we employed two datasets of large real networks used in [8], which exhibit many of the key features of social networks. The first one is a trackback network of Japanese blogs. The network data were collected by tracing the trackbacks from one blog in the site *goo*[3] in May, 2005. We refer to this network data as the blog network. The blog network was a strongly-connected bidirectional network, where a link created by a trackback was regarded as a bidirectional link since blog authors establish mutual communications by putting trackbacks on each other's blogs. The blog network had $12,047$ nodes and $79,920$ directed links. The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia. We refer to this network data as the Wikipedia network. The Wikipedia network was also a strongly-connected bidirectional network, and had $9,481$ nodes and $245,044$ directed links.

We determined the values of r and κ of the two models which we assumed to be true in the following way. In the AsIC model, we calculated the mean out-degree $\bar{d}$ and set two different values of κ in reference to $1/\bar{d}$, one smaller than $1/\bar{d}$ according to [7] and the other larger than $1/\bar{d}$ to see how a different value affects the result. Since the values of $\bar{d}$ were about 6.63 and 25.85 for the blog and the Wikipedia networks, respectively, the corresponding values of $1/\bar{d}$ were about 0.15 and 0.03. Thus, we decided to set $\kappa = 0.1$ and 0.3 for the blog network and $\kappa = 0.03$ and 0.09 for the Wikipedia network as the true values. As for the time-delay parameter r, we simply decided to set it to 1.0 because changing r is equivalent to changing the time scale accordingly. In the AsLT model, we only chose one value for κ. This is because we found that the information does not reach out far in the AsLT model and we needed to set a large value for κ to realize a decent diffusion. A value of 0.9 was a proper choice for κ. The time-delay parameter was set to $r = 1.0$, same as for the AsIC model.

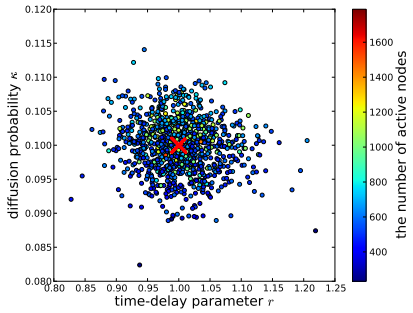
5.3 Experimental results

blog network under the AsIC model Figure 1 is the results of blog network under the AsIC model for the parameters $\kappa = 0.1$ and $r = 1.0$ (proc. 1). Figure 1(a) plots individ-

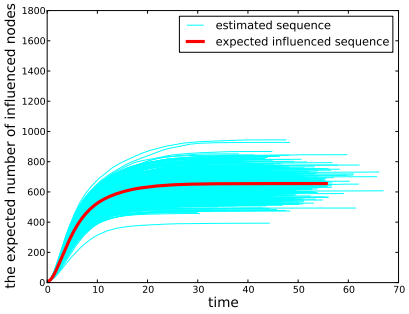
[3] `http://blog.goo.ne.jp/`



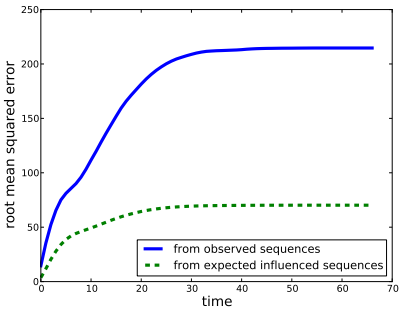
(a) Curves for generated sequences



(b) Parameter estimation results



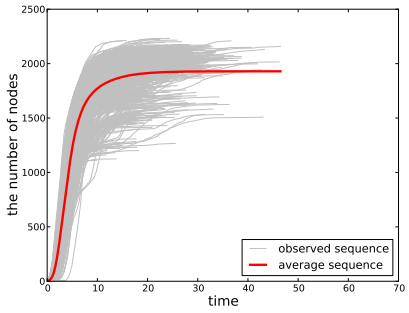
(c) Curves from estimated parameters



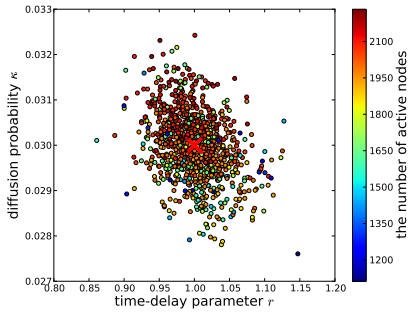
(d) Accuracy comparison

Fig. 1: The result set of blog network under the AsIC model ($\kappa^* = 0.1$)

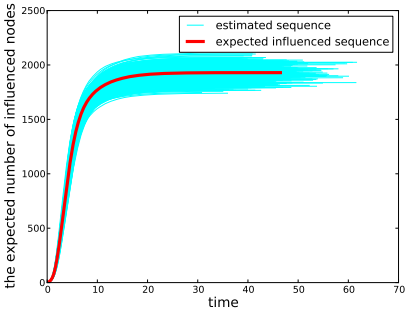
ual sequence data when the diffusion simulation was repeated $N = 1000$ times starting from the same initial source node (proc. 2). The horizontal axis is the time and the vertical axis is the number of active nodes. As shown in the figure, we observe a wide variety of influence curves with respect to time (dipicted in grey) due to the stochastic nature of the AsIC model. Here our task is to estimate the expected influence curve (depicted in red (black)), which is approximated by the empirical mean of the 1000 gray curves (proc. 3). Figure 1(b) is to show that it is possible to estimate the parameters of the AsIC model, i.e. time-delay parameter r and diffusion probability κ even from a single diffusion sequence. There are 1000 dots and each dot is the esitmated results (r, κ) from the corresponding sequence (proc. 4). We observe that the parameter estimation results are scatterd around the true values $(r^*, \kappa^*) = (1.0, 0.1)$, which were used to generate each sequence. The color (greyness) in the bar on the right indicates the lengh of the sequence, and the results are not very sensitive to the length unless it is very short. Figure 1(c) shows the estimated influence curves (depicted in cyan (grey)), each of which is obtained by performing simulation $K = 100$ times from the correspondng initial source node using the AsIC model with the same parameters learned from the corresponding original diffusion sequence. The target expected influence curve is the same as in



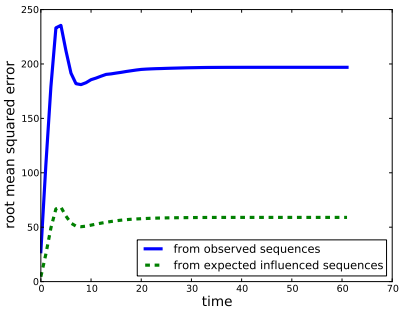
(a) Curves for generated sequences



(b) Parameter estimation results



(c) Curves from estimated parameters

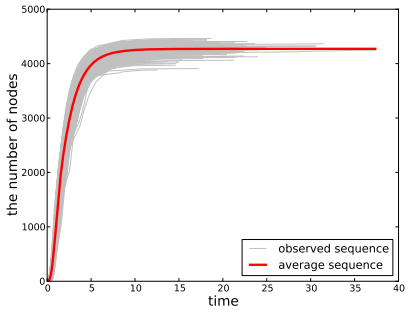


(d) Accuracy comparison

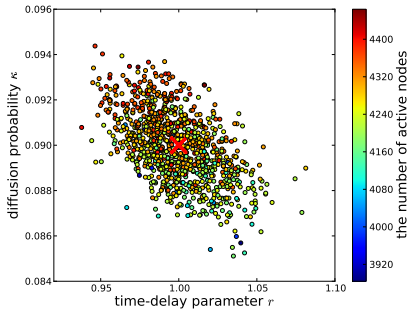
Fig. 2: The result set of Wikipedia network under the AsIC model ($\kappa^* = 0.03$)

Figure 1(a). Figure 1(d) shows the RMSE (Root Mean Squared Error) curves for both the original influence $\varphi(t; v_0, d_n)$ (Figure 1(a)) and the estimated influence $\sigma(t; v_0, d_n)$ (Figure 1(c)) with respect to the target influence (proc. 5). As shown, we observe that the RMSE for the estimated curve is much smaller (less than 1/3) than the one for the original one. Thus, we can say that the estimated influence curve is much closer to the expected influence curve than the original curve. Similar result is obtained for the case of $\kappa^* = 0.3$.

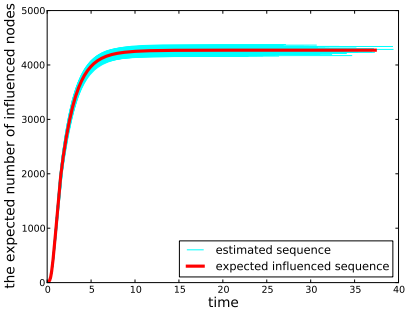
Wikipedia network under the AsIC model Figures 2 and 3 are the results of Wikipedia network under the AsIC model for $\kappa^* = 0.03$ and $\kappa^* = 0.09$, respectively. In both cases, the RMSE for the estimated curve is much smaller (about 1/4 for $\kappa^* = 0.03$ and about 1/2 for $\kappa^* = 0.09$) in the proposed method. The results for $\kappa^* = 0.03$ is similar to the results of blog network except that the shape of the RMSE curve is different. However, the results for $\kappa^* = 0.09$ reveal different behaviors. When the diffusion probability is large, the information propagates far enough and individual sequence becomes similar to each other. Note that the number of nodes is almost doubled. The accuracy becomes better accordingly, especially for the original influence $\varphi(t; v_0, d_n)$. In general the proposed



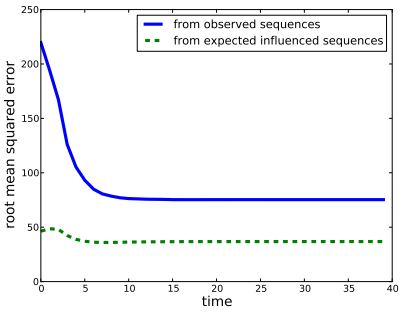
(a) Curves for generated sequences



(b) Parameter estimation results



(c) Curves from estimated parameters



(d) Accuracy comparison

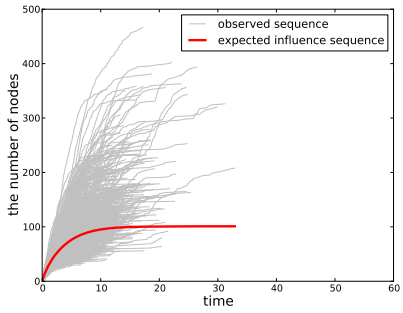
Fig. 3: The result set of Wikipedia network under the AsIC model ($\kappa^* = 0.09$)

method is more effective when the diffusion probability is small and the observation sequences are diversified.

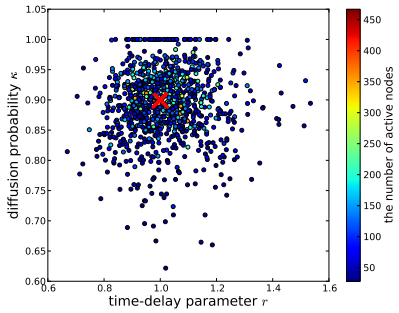
blog network under the AsLT model Figure 4 shows the results of blog network under the AsLT model for $\kappa^* = 0.9$. Unlike the AsIC model, the information does not spread far and wide and the sequences are short. Accordingly the number of active nodes are much smaller (less than 500) and the errors in the parameter estimation are larger than the AsIC model. But still, we can say that the parameters are estimated reasonably well and the RMSE is much smaller (about 1/3) in the proposed method. Similar results are obtained for Wikipedia network.

5.4 Visual Analyses

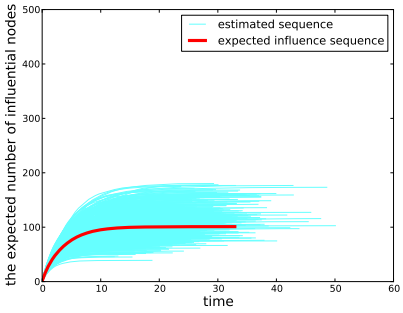
We saw that observation sequences are diverse in general due to the stochastic nature of the diffusion process. The differences in diffusion patterns are best understood by visualizing the active nodes. Figure 5 visualizes two extreme diffusion patterns for blog



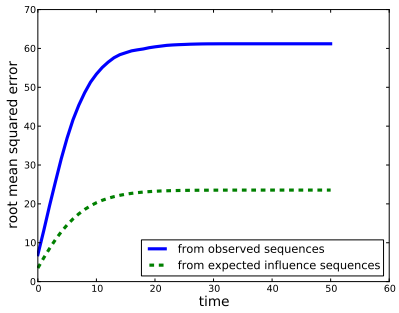
(a) Curves for generated sequences



(b) Parameter estimation results



(c) Curves from estimated parameters



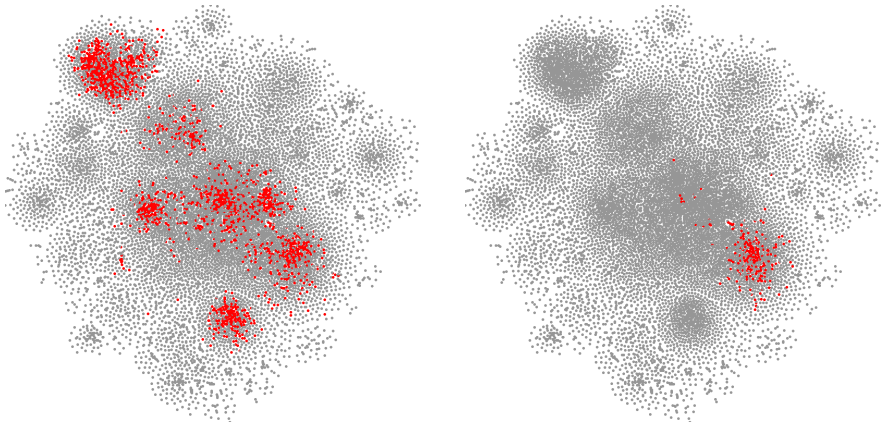
(d) Accuracy comparison

Fig. 4: The result set of blog network under the AsLT model ($\kappa^* = 0.9$)

network of Figure 2 by using Cross-entropy method [15]. The red dots indicate active nodes and the gray dots non-active nodes. Figure 5(a) is the pattern for the longest sequence and Figure 5(b) is the one for the shortest sequence. We observe that dots are not uniformly distributed but have some dense reagions forming communities. In Figure 5(a) the information diffuses across many communities and spread widely, whereas in Figure 5(b) it is trapped within the same community of the initial source node and does not spread. Consequently, the number of active nodes in Figure 5(a) is 1,789 and that in Figure 5(b) is only 220. Simialr result is also observed in Wikipedia network.

6 Discussion

We note that the analysis we showed in this paper is the simplest case where κ and r take a single value each for all the links in E. However, the method is very general. In a more realistic setting we can divide E into subsets $E_1, E_2, ..., E_N$ and assign a different value κ_n and r_n for all the links in each E_n. For example, we may divide the nodes into two groups: those that strongly influence others and those not, or we may divide the



(a) Visualization of high diffusion result (b) Visualization of low diffusion result

Fig. 5: Visualization of blog network

nodes into another two groups: those that are easily influenced by others and those not. We can further divide the nodes into multiple groups. In this setting we learn κ_n and r_n for $n = 1, 2, ..., N$ from a single observation sequence.

We aimed to estimate the expected influence curve assuming two different information diffusion models in this paper but the framework of the proposed method can be applied to other models as well as other measures. For example, if we are interested in how different opinions spread [16], we can use the Voter model and estimate the expected opiniton share curve under this framework. Which measure and model to use depends on the problem we want to solve and the evaluation must be based on a task-specific performance measure.

7 Conclusion

One of the challenges of social network analysis is to estimate the expected influence degree with respect to time (expected influence curve). Because of the stochastic nature of information diffusion, a single observation sequence is not reliable to use as an approximation of this curve. We proposed a novel method to estimate the expected influence curve with good accuracy from a single observed information diffusion sequence assuming two types of information diffusion models: the asynchronous independent cascade (AsIC) model and the asynchronous linear threshold (AsLT). The method first learns the model parameters from a single observation sequence and next use the learned model to estimate the expected influence curve. We showed that parameter learning from a single sequence is feasible and practical, and the estimated influence curve is much more accurate than using the observed sequence as its approximation by extensive experiments using two real world networks.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Physical Review E **66** (2002) 035101
2. Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45** (2003) 167–256
3. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. SIGKDD Explorations **6** (2004) 43–52
4. Domingos, P.: Mining social networks for viral marketing. IEEE Intelligent Systems **20** (2005) 80–82
5. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06). (2006) 228–237
6. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters **12** (2001) 211–223
7. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003). (2003) 137–146
8. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data **3** (2009) 9:1–9:23
9. Watts, D.J.: A simple model of global cascades on random networks. Proceedings of National Academy of Science, USA **99** (2002) 5766–5771
10. Watts, D.J., Dodds, P.S.: Influence, networks, and public opinion formation. Journal of Consumer Research **34** (2007) 441–458
11. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07). (2007) 1371–1376
12. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Finding influential nodes in a social network from information diffusion data. In: Proceedings of the International Workshop on Social Computing and Behavioral Modeling (SBP09). (2009) 138–145
13. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Learning continuous-time information diffusion model for social behavioral data analysis. In: Proceedings of the 1st Asian Conference on Machine Learning (ACML2009). (2009) 322–337
14. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Behavioral analyses of information diffusion models by observed data of social network. In: Proceedings of the the 2010 International Conference on Social Computing, Behavioral Modeling, and Prediction (SBP 2010). (2010) 149–158
15. Yamada, T., Saito, K., Ueda, N.: Cross-entropy directed embedding of network data. In: Proceedings of the 20th International Conference on Machine Learning (ICML03). (2003) 832–839
16. Kimura, M., Saito, K., Motoda, H., Ohara, K.: Learning to predict opinion share in social networks. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-10). (2010)

Finding Relation between PageRank and Voter Model

Takayasu Fushimi[1], Kazumi Saito[1], Masahiro Kimura[2], Hiroshi Motoda[3], and Kouzou Ohara[4]

[1] Graduate School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
{j09118,k-saito}@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
Otsu, Shiga 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[3] Institute of Scientific and Industrial Research, Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp
[4] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp

Abstract. Estimating influence of a node is an important problem in social network analyses. We address this problem in a particular class of model for opinion propagation in which a node adopts its opinion based on not only its direct neighbors but also the average opinion share over the whole network, which we call an extended Voter Model with uniform adoption (VM). We found a similarity of this model with the well known PageRank (PR) and explored the relationships between the two. Since the uniform adoption implies the random opinion adoption of all nodes in the network, it corresponds to the random surfer jump of PR. For an undirected network, both VM and PR give the same ranking score vector because the adjacency matrix is symmetric, but for a directed network, the score vector is different for both because the adjacency matrix is asymmetric. We investigated the effect of the uniform adoption probability on ranking and how the ranking correlation between VM and PR changes using four real world social networks. The results indicate that there is little correlation between VM and PR when the uniform adoption probability is small but the correlation becomes larger when both the uniform adoption and the random surfer jump probabilities become larger. We identified that the recommended value for the uniform adoption probability is to be around 0.25 to obtain a stable solution.

1 Introduction

Recent technological innovation in the web such as blogosphere and knowledge/media-sharing sites is remarkable, which makes it possible to form various kinds of large social networks, through which behaviors, ideas and opinions can spread. Thus, substantial attention has been directed to investigating the spread of influence in these networks [11, 3, 16]. The representative problem is the influence maximization problem, that is, the problem of finding a limited number of influential nodes that are effective for the

spread of information through the network and new algorithmic approaches have been proposed under different model assumptions, e.g. descriptive probabilistic interaction models [4, 14], and basic diffusion models such as independent cascade (IC) model and the linear threshold (LT) model [7, 8, 18]. This problem has good applications in sociology and "viral marketing" [1].

Another line of work on the spread of influence is opinion share analyses, i.e. how people changes their opinions, how each opinion propagates and what the final opinion share is, etc. A good model for opinion diffusion would be a voter model [12, 15]. It is one of the most basic stochastic process model, and has the same key property with the linear threshold model that a node decision is influenced by its neighbors' decision, i.e. a person changes its opinion by the opinions of its neighbors. The basic voter model is defined on an undirected network with self-loop and each node initially holds one of K opinions, and adopts the opinion of a randomly chosen neighbor at each subsequent discrete time-step.

Even-Dar and Shapira [5] investigated the influence maximization problem (maximizing the spread of the opinion that supports a new technology) under the basic voter model with two ($K = 2$) opinions (one in favor of the new technology and the other against it) at a given target time T. They showed that the most natural heuristic solution, which picks the nodes in the network with the highest degree, is indeed the optimal solution, under the condition that all nodes have the same cost.

We propose a new model for the spread of opinions. Each person has a different influence on the other person and the person to person relation is directional. A person not only changes its opinion by its direct neighbors but also considers the overall opinion distributions of the whole society. The new model incorporates these factors and we call this model as an extended Voter Model with uniform adoption. Here we note that the new model has a strong similarity to the well known PageRank [2, 10] which is an algorithm to rank Web pages. Since the uniform adoption can be viewed as random opinion adoption of all nodes in the network, it is equivalent to the random surfer jump of PageRank.

We mathematically derive the ranking vector of the new Voter Model and compare it with that of PageRank, and explore how the two models are related by a series of extensive experiments using four real world social networks. Especially we investigate the effects of the uniform adoption probability on node ranking and how the ranking of the new Voter Model and PageRank are correlated to each other with this probability. The ranking of the new Voter Model becomes the same as that of PageRank if we assume that the network is undirectional, but since both our new model and PageRank use directional network, the ranking results are not the same. The results indicate that the correlation varies with the uniform adoption probability. There is little correlation between the extended Voter Model and PageRank when the uniform adoption probability is small and the high ranked nodes are different, but the correlation becomes larger when both the uniform adoption and the random surfer jump probabilities become larger. We found that the ranking becomes stable for the uniform adoption probability in the range of 0.15 and 0.35 and the self correlation within the extended Voter Model is high in this region, which is consistent with the report that the recommended value for the random surfer jump is 0.15.

The paper is organized as follows. We briefly explain the standard Voter Model and revisit PageRank in sections 2 and 3, respectively. Then we explain our new Voter Model, the extended Voter Model with uniform adoption, in section 4. Experimental results that describe various correlation results are detailed in section 5. Finally we summarize our conclusion in section 6.

2 Voter Model

In this section, according to the work [5], we first consider the diffusion of opinions in a social network represented by an undirected (bidirectional) graph $G = (V, E)$ with self-loops. Here, V and E ($\subset V \times V$) are the sets of all the nodes and links in the network, respectively. For a node $v \in V$, let $\Gamma(v)$ denote the set of neighbors of v in G, that is, $\Gamma(v) = \{u \in V; (u, v) \in E\}$. Note that $v \in \Gamma(v)$.

According to the work [5], we recall the definition of the basic voter model with two opinions on network G. In the voter model, each node of G is endowed with two states; opinions 1 and 2. The opinions are initially assigned to all the nodes in G, and the evolution process unfolds in discrete time-steps $t = 1, 2, 3, \cdots$ as follows: At each time-step t, each node v picks a random neighbor u and adopts the opinion that u holds at time-step $t - 1$.

More formally, let $f_t : V \to \{1, 2\}$ denote the opinion distribution at time-step t, where $f_t(v)$ stands for the opinion of node v at time-step t. Then, $f_0 : V \to \{1, 2\}$ is the initial opinion distribution, and $f_t : V \to \{1, 2\}$ is inductively defined as follows: For any $v \in V$,

$$\begin{cases} f_t(v) = 1, \text{ with probability } \frac{n_{t-1}(1,v)}{n_{t-1}(1,v) + n_{t-1}(2,v)}, \\ f_t(v) = 2, \text{ with probability } \frac{n_{t-1}(2,v)}{n_{t-1}(1,v) + n_{t-1}(2,v)}, \end{cases}$$

where $n_t(k, v)$ is the number of v's neighbors that hold opinion k at time-step t for $k = 1, 2$.

3 PageRank Revisited

We revisit PageRank [2, 10]. For a given Web network (directed graph), we identify each node with a unique integer from 1 to $|V|$. Then we can define the adjacency matrix $\boldsymbol{A} \in \{0, 1\}^{|V|\times|V|}$ by setting $a(u, v) = 1$ if $(u, v) \in E$; otherwise $a(u, v) = 0$. A node can be self-looped, in which case $a(u, u) = 1$. For each node $v \in V$, let $F(v)$ and $B(v)$ denote the set of child nodes of v and the set of parent nodes of v, respectively, $F(v) = \{w \in V;\ (v, w) \in E\}$, $B(v) = \{u \in V;\ (u, v) \in E\}$. Note that $v \in F(v)$ and $v \in B(v)$ for node v with a self-loop.

Then we can consider the row-stochastic transition matrix $\boldsymbol{P}$, each element of which is defined by $p(u, v) = a(u, v)/|F(u)|$ if $|F(u)| > 0$; otherwise $p(u, v) = z(v)$, where $\boldsymbol{z}$ is some probability distribution over pages, i.e., $z(v) \geq 0$ and $\sum_{v\in V} z(v) = 1$. This model means that from dangling Web pages without out-links ($F(u) = \emptyset$), a random surfer jumps to page v with probability $z(v)$. The vector $\boldsymbol{z}$ is referred to as a personalized vector because we can define $\boldsymbol{z}$ according to user's preference.

Let $\boldsymbol{y}$ denote a vector representing PageRank scores over pages, where $y(v) \geq 0$ and $\sum_{v \in V} y(v) = 1$. Then using an iteration-step parameter t, PageRank vector $\boldsymbol{y}$ is defined as a limiting solution of the following iterative process,

$$\boldsymbol{y}_t^T = \boldsymbol{y}_{t-1}^T \left((1-\beta)\boldsymbol{P} + \beta \boldsymbol{e}\boldsymbol{z}^T \right) = (1-\beta)\boldsymbol{y}_{t-1}^T \boldsymbol{P} + \beta \boldsymbol{z}^T, \tag{1}$$

where $\boldsymbol{a}^T$ stands for a transposed vector of $\boldsymbol{a}$ and $\boldsymbol{e} = (1, \cdots, 1)^T$. In the Equation (1), β is referred to as the uniform jump probability. This model means that with the probability β, a random surfer also jumps to some page according to the probability distribution $\boldsymbol{z}$. The matrix $((1-\beta)\boldsymbol{P} + \beta \boldsymbol{e}\boldsymbol{z}^T)$ is referred to as a Google matrix. The standard PageRank method calculates its solution by directly iterating Equation (1), after initializing $\boldsymbol{y}_0$ adequately. One measure to evaluate its convergence is defined by

$$\|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_{L1} \equiv \sum_{v \in V} |y_t(v) - y_{t-1}(v)|. \tag{2}$$

Note that any initial vector $\boldsymbol{y}_0$ can give almost the same PageRank scores if it makes Equation (2) almost zero because the unique solution of Equation (1) is guaranteed.

4 Voter Model with uniform adoption

We propose an extended Voter Model with uniform adoption on a directed graph $G = (V, E)$ with self-loops for K opinions. Let $m_t(k, v)$ be the number of v's parents that hold opinion k at time-step t for $k = 1, 2, ..., K$. In addition, just like the personalized vector employed in PageRank, we introduce some probability distribution $\boldsymbol{z}$ over nodes. Let $m_t(k)$ be the weighted share of opinion k at time-step t given by

$$m_t(k) = \sum_{\{v \in V;\, f_t(v) = k\}} z(v), \tag{3}$$

then $f_t : V \to \{1, 2, ..., K\}$ is inductively defined as follows, given an initial opinion distribution $f_0 : V \to \{1, 2, ..., K\}$. For any $v \in V$,

$$f_t(v) = k, \quad \text{with probability } (1-\alpha)\frac{m_{t-1}(k, v)}{\sum_{k=1}^{k=K} m_{t-1}(k, v)} + \alpha m_{t-1}(k). \tag{4}$$

This model indicates that the opinion of each node $v \in V$ is influenced by its parents nodes $B(v)$ with probability $(1-\alpha)$ and by any other node $u \in V$ with probability α according to $\boldsymbol{z}$. Hereafter, α is referred to as the uniform adoption probability and the extended Voter Model with uniform adoption is referred to as VM for short[2].

Now we consider estimating the expected influence degree of node $u \in V$, which is defined as the expected number of nodes influenced by u's initial opinion $f_0(u)$. Note that the following definition does not depend on which opinion u holds initially. We denote the expected influence degree of node u at time-step t by $x_t(u)$. Let $\boldsymbol{h}_u \in \{0, 1\}^{|V|}$ be a vector whose u-th element is 1 and other elements are 0, and $\boldsymbol{Q}$ the column-stochastic

[2] We call it as the extended VM when we have to make distinction from the standard VM.

transition matrix, each element of which is defined by $q(u,v) = a(u,v)/|B(v)|$. Here note that $B(v) \neq \emptyset$ for any node $v \in V$ because of the existence of self-loop. From the definition of our model, we can calculate $x_1(u)$ as follows.

$$x_1(u) = (1-\alpha) \sum_{v \in F(u)} |B(v)|^{-1} + |V|\alpha z(u) = \boldsymbol{h}_u^T \left((1-\alpha)\boldsymbol{Q} + \alpha \boldsymbol{z}\boldsymbol{e}^T\right)\boldsymbol{e}. \tag{5}$$

Each element of the vector $\boldsymbol{h}_u^T((1-\alpha)\boldsymbol{Q}+\alpha\boldsymbol{z}\boldsymbol{e}^T)$ is the probability that the corresponding node v is influenced by the node u with one time-step. Thus from the independence property of the opinion diffusion process, we can calculate $x_t(u)$ as follows.

$$x_t(u) = \boldsymbol{h}_u^T \left((1-\alpha)\boldsymbol{Q} + \alpha \boldsymbol{z}\boldsymbol{e}^T\right)^t \boldsymbol{e}. \tag{6}$$

Here since the vector $\boldsymbol{h}_u$ works for selecting the u-th element, we can obtain the vector consisting of the expected influence degree at time-step t as follows:

$$\boldsymbol{x}_t = \left((1-\alpha)\boldsymbol{Q} + \alpha \boldsymbol{z}\boldsymbol{e}^T\right)^t \boldsymbol{e} = \left((1-\alpha)\boldsymbol{Q} + \alpha \boldsymbol{z}\boldsymbol{e}^T\right)^{t-1} \boldsymbol{x}_{t-1} \tag{7}$$

Moreover, since $((1-\alpha)\boldsymbol{Q}+\alpha\boldsymbol{z}\boldsymbol{e}^T)$ becomes the column-stochastic transition matrix, we can consider a stationary vector defined by $\boldsymbol{x} = \lim_{t\to\infty} \boldsymbol{x}_t$.

For the sake of technical convenience, we perform scaling to the vector $\boldsymbol{x}$ defined by $\boldsymbol{x} \leftarrow \boldsymbol{x}/|V|$. Then, similarly to PageRank calculation process defined in Equation (1), we can obtain the expected influence vector at time-step t as follows after initializing vector to $\boldsymbol{x}_0 = \boldsymbol{e}/|V|$:

$$\boldsymbol{x}_t = \left((1-\alpha)\boldsymbol{Q} + \alpha \boldsymbol{z}\boldsymbol{e}^T\right)\boldsymbol{x}_{t-1} = (1-\alpha)\boldsymbol{Q}\boldsymbol{x}_{t-1} + \alpha\boldsymbol{z}. \tag{8}$$

We can employ the same convergence measure defined by Equation (2), just by replacing the vector $\boldsymbol{y}$ with $\boldsymbol{x}$. Here, we note that in case of undirected networks with self-loops Equations (1) and (8) become completely equivalent since there exist no dangling nodes. Note also that in this case, our extended VM reduces to the standard VM by setting $\alpha = 0$. On the other hand, in case of directed networks with self-loops, Equations (1) and (8) give different vector sequences, and we empirically evaluate their differences with special emphasis on their stationary vectors.

5 Experiments

In this section, we evaluate the effects of the uniform adoption probability α and those of community structure in our VM, and examine the relation between VM and PR by extensive experiments using four real networks.

5.1 Experimental settings

In our experiments, we employ the Pearson correlation coefficients as our basic evaluation measure. For the sake of convenience, we recall its definition: given two vectors, $\boldsymbol{x}$ and $\boldsymbol{y}$, the correlation coefficient $C(\boldsymbol{x},\boldsymbol{y})$ is defined as follows.

$$C(\boldsymbol{x},\boldsymbol{y}) = \frac{(\boldsymbol{x}-\bar{x}\boldsymbol{e})^T(\boldsymbol{y}-\bar{y}\boldsymbol{e})}{\sqrt{(\boldsymbol{x}-\bar{x}\boldsymbol{e})^T(\boldsymbol{x}-\bar{x}\boldsymbol{e})}\sqrt{(\boldsymbol{y}-\bar{y}\boldsymbol{e})^T(\boldsymbol{y}-\bar{y}\boldsymbol{e})}}, \tag{9}$$

where $\bar{x}$ and $\bar{y}$ stand for the average element values of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and recall that $\boldsymbol{e}$ is a vector defined by $\boldsymbol{e} = (1, \cdots, 1)^T$.

As mentioned earlier, we focus on evaluating the vectors of the expected influence degree, each of which is the stationary vector defined as a limiting solution of Equation (8) in VM. In our experiments, the personalized vector $\boldsymbol{z}$ is set to uniform one, i.e., $\boldsymbol{z} = (1/|V|, \cdots, 1/|V|)^T$. Based on Equation (2), the convergence criterion to obtain the stationary vectors is set to $\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_{L1} < 10^{-12}$ in case of VM, and $\|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_{L1} < 10^{-12}$ in case of PR.

Our evaluation consists of two series of experiments. In the first series of experiments, we evaluate the effects of the uniform adoption probability on the expected influence degree. In the second series of experiments, we evaluate the effects of network's community structure on the expected influence degree. Now, we explain our method of rewiring the originally observed network to change its community structure. The rewired network is constructed just by randomly rewiring links of the original network according to some probability p without changing the degree of each node [13]. More specifically, by arbitrarily ordering all links except for self-loops in a given original network, we can prepare a link list $L_E = (e_1, \cdots, e_{|E|})$. Recall that each directed link consists of an ordered pair of *from*-part and *to*-part nodes, i.e., $e = (u, v)$. From the list L_E, we can produce two node lists, i.e., the *from*-part node list L_F and the *to*-part node list L_T. We assign the out- (or in-) degree for each node v appearing in L_F (or L_T). Thus, by swapping two elements of the node list L_T with the probability p so as not to produce multiple-links, we can obtain a partially reordered node list L'_T. Then, by concatenating L'_T with the other node list L_F, we can produce a link list for a rewired network. Namely, let L'_T be a shuffled node list, and we denote the i-th order element of a list L by $L(i)$; then the link list of the rewired network is $L'_E = ((L_F(1), L'_T(1)), \cdots, (L_F(|E|), L'_T(|E|)))$.

5.2 Network Data

In our experiments, we employed four sets of real networks, which exhibit many of the key features of social networks. Below we describe the details of these network data.

The first one is a reader network of "Ameba"[3] that is a Japanese blog service site. Blogs are personal on-line diaries managed by easy-to-use software packages, and have rapidly spread through the World Wide Web [6]. Each blog of "Ameba" can have the *reader list* that consists of the hyperlinks to the blogs of the reader bloggers for her blog. Here, a reader link from blog X to blog Y is generated when blog Y registers blog X as her favorite blog. Thus, a reader network can be regarded as a social network. We crawled the reader lists of $117,374$ blogs of the Ameba blog service site in June 2006, and collected a large connected network. This network had $56,604$ nodes and $1,071,080$ directed links. We refer to this network as the Ameblo network.

Second one is a trackback network of blogs used in [8]. Bloggers discuss various topics by using trackbacks. Thus, a piece of information can propagate from one blogger to another blogger through a trackback. We exploited the blog "Theme salon of blogs" in the site "goo" [4], where a blogger can recruit trackbacks of other bloggers

[3] http://www.ameba.jp/

[4] http://blog.goo.ne.jp/usertheme/

by registering an interesting theme. By tracing up to ten steps back in the trackbacks from the blog of the theme "JR Fukuchiyama Line Derailment Collision", we collected a large connected trackback network in May, 2005. The resulting network had $12,047$ nodes and $79,920$ directed links. We refer to this network data as the Blog network.

The third one is a fan network of "@cosme"[5] that is a Japanese word-of-mouth communication site for cosmetics. Each user page of "@cosme" can have *fan links*. Here, a fan link from user X to user Y is generated when user Y registers user X as her favorite user. Thus, a fan network can be regarded as a social network. We traced up to ten steps in the fan links from a randomly chosen user in December 2009, and collected a large connected network [6]. This network had $45,024$ nodes and $546,930$ directed links. We refer to this network as the Cosme network.

Last we employed a network derived from the Enron Email Dataset [9]. We first extracted the email addresses that appeared in the Enron Email Dataset as senders and recipients. We regarded each email address as a node, and constructed a directed network obtained by linking two email addresses u and v if u sent an email to v Next, we extracted its maximal strongly connected component. We refer to this strongly connected bidirectional network as the Enron network. This network had $4,254$ nodes and $44,314$ directed links. We refer to this dataset as the Enron network dataset.

Table 1: Basic statistics of networks.

network	$\|V\|$	$\|E\|$	$C(\boldsymbol{B}, \boldsymbol{F})$
Ameblo	56,604	1,071,080	0.61350
Blog	12,047	79,920	0.74377
Cosme	45,024	546,930	0.51940
Enron	19,654	377,612	0.54929

Table 1 shows the basic statistics of the Ameblo, Blog, Cosme and Enron networks. Here, $C(\boldsymbol{B}, \boldsymbol{F})$ denotes the Peason correlation coefficients between the in-degree vector $\boldsymbol{B}$, each element of which is $|B(v)|$, and the out-degree vector $\boldsymbol{F}$, each element of which is $|F(v)|$. From this table, we consider that each network has an intrinsic characteristics as a directed network because $C(\boldsymbol{B}, \boldsymbol{F})$ is reasonably smaller than 1.

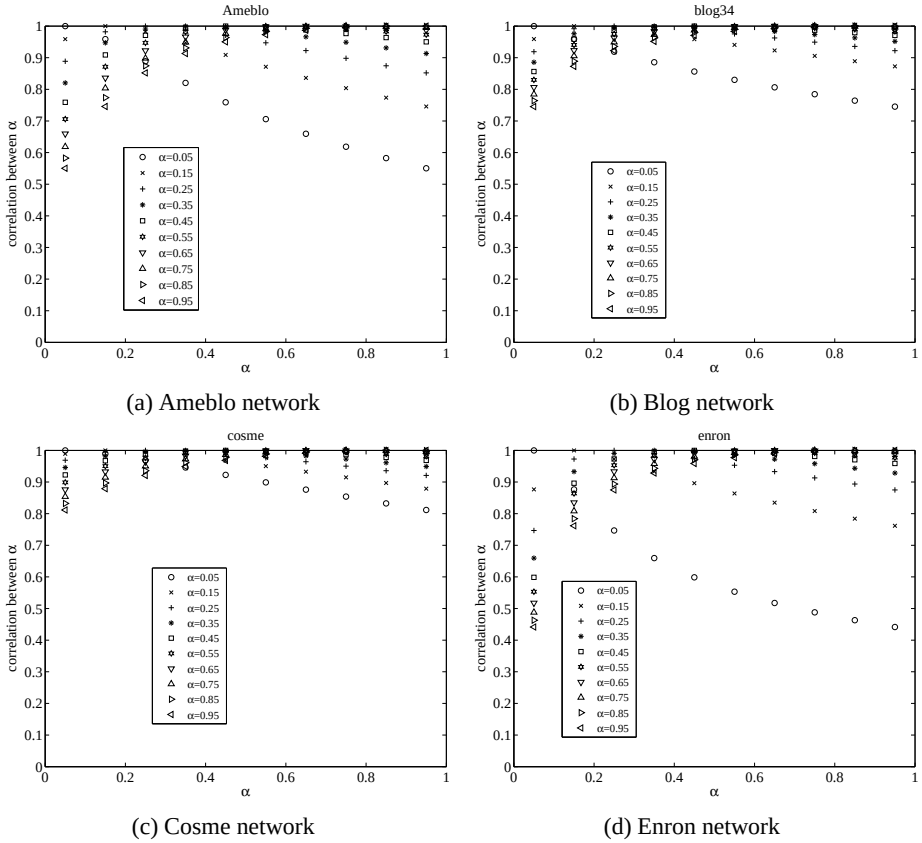
5.3 Effects of uniform adoption probability

As the first series of experiments, we evaluated the effects of the uniform adoption probability change on the expected influence degree. Here, let $\boldsymbol{x}(\alpha)$ be the statinary vector defined as a limmiting solution of Equation (8) for VM with α. In order to evaluate the effects of different uniform adoption probabilities, we calculated the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha'))$ with respect to each pair of the uniform adoption probabilities, α

[5] `http://www.cosme.net/`

[6] We tried this collection procedure twice, and compared the resulting networks. Then, we found that they overlapped 99.5%.

and α' (self correlation). In Fig.1, we plot $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha'))$ with respect to α, where each result with different α' is shown by a different maker. Here we changed both the values of α and α' from 0.05 to 0.95 with an increment of 0.1.



(a) Ameblo network (b) Blog network

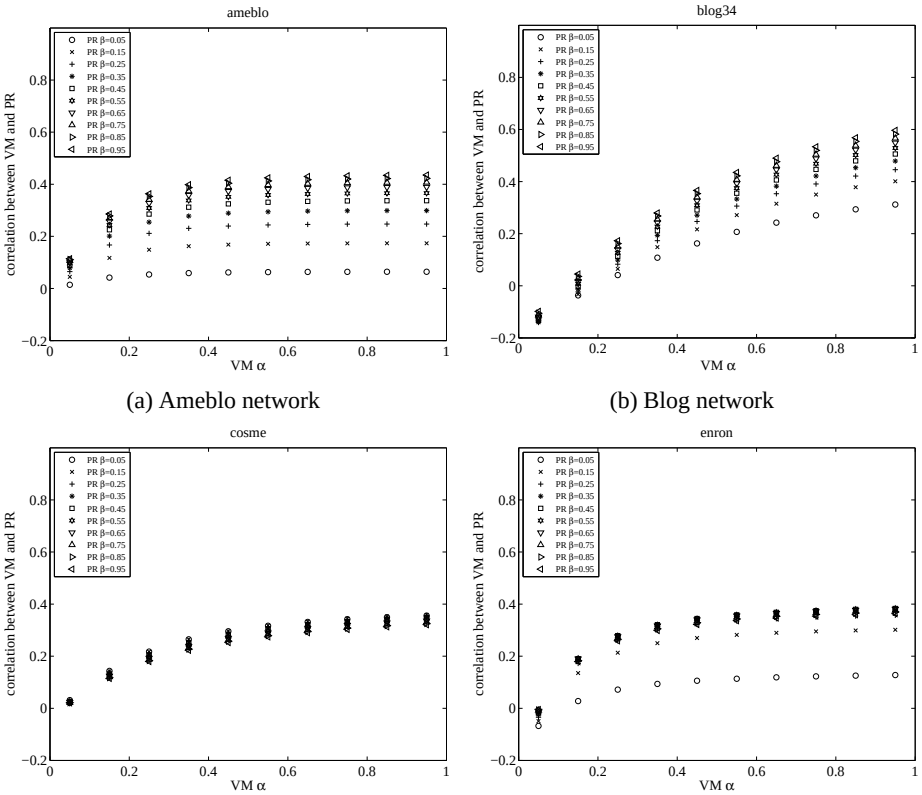
(c) Cosme network (d) Enron network

Fig. 1: The correlation coefficient between VMs with different α

From Fig.1, we can observe the following smilar characteristics of VM for all of the four networks. First, the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha'))$ for any pair of α and α' are relatively high. Second, $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha'))$ in the range of $0.15 \leq \alpha \leq 0.35$ shows especially high values regardless of α. This suggests that we can recommend to employ this range of α because this would give a stable (and thus, representative) value of the expected influence degree for VM. Incidentally, it is reported that the uniform jump probability β in PR is frequently used at $\beta = 0.15$ [2, 10]. Third, we can see that when $\alpha = 0.05$, $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha'))$ decreases almost linearly as α' increases, while it decreases very little for small values of α' and only modestly for large values of α' when $\alpha = 0.95$.

Similarly to the above, let $\boldsymbol{y}(\beta)$ be the statinary vector defined as a limmiting solution of Equation (1) for PR with β. In order to examine the relation between VM and PR,

we calculated the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\beta))$ with respect to each pair of the uniform adoption probability α and the uniform jump probability β. In Fig.2, we plot $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\beta))$ with respect to α, where each result with different β is shown by a different maker. Here we also changed the values of β from 0.05 to 0.95 with an increment of 0.1.



(a) Ameblo network (b) Blog network

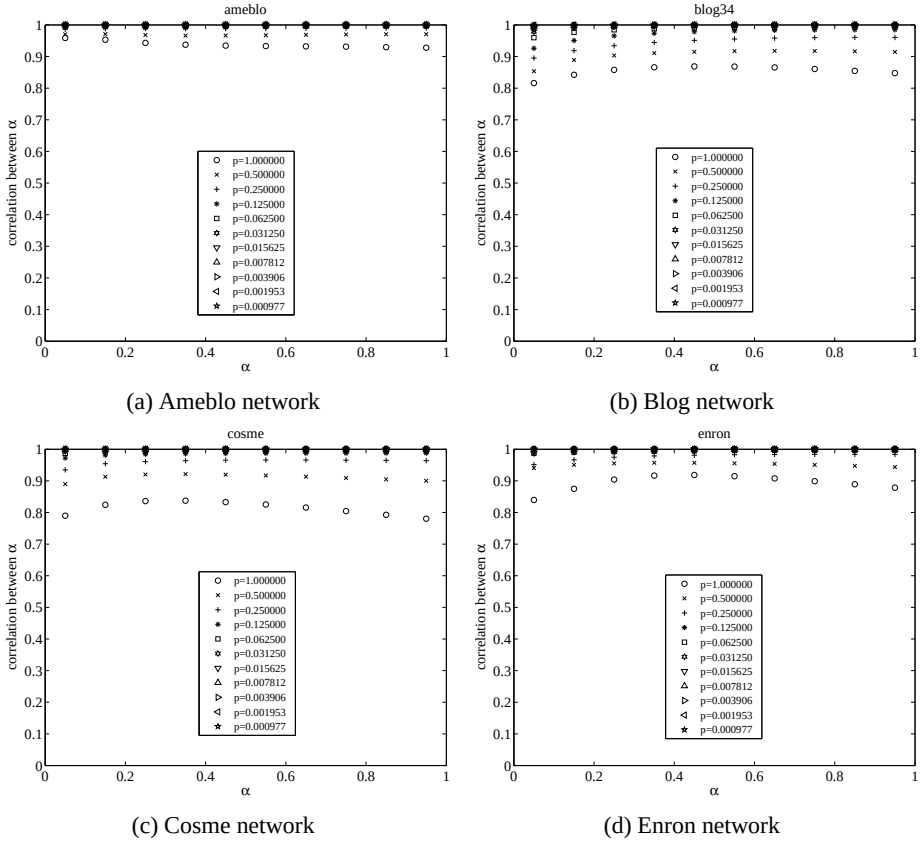
(c) Cosme network (d) Enron network

Fig. 2: The correlation coefficient between VM with α and PR with β

From Fig.2, we can observe the following similar relationships between VM and PR for all of the four networks. First, when α is small, there exists almost no correlation between the expected influence degree and the PageRank score. Second, for any β, $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\beta))$ generally increses as α increases, although their rates of increase depend on β as well as the network. Third, the maximum values of $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\beta))$ are attained at $\alpha = 0.95$. Incidentally, these maximum values are somewhat smaller than the correlation coefficients between in- and out-degree vectors, $C(\boldsymbol{B}, \boldsymbol{F})$, shown in Table 1, but their relative values are consistent between the two.

5.4 Effects of community structure

As the second series of experiments, we evaluated the effects of the community structure change on the expected influence degree. To this end, we constructed the 10 rewired networks from each of the original four networks using the rewiring probability $p = 2^{-k}$ ($k = 1, 2, \cdots, 10$) so that each network has a different community structure with different degree from the original one's (see the rewiring method in Section 5.1). Now, let $\boldsymbol{x}(\alpha, p)$ be the statinary vector calculated from the network rewired with probability p for VM with α. In order to evaluate the effects of different community structure, we calculated the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha, p))$ with respect to each pair of the uniform adoption probability α and the rewiring probability p. In Fig.3, we plot $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha, p))$ with respect to α, where each result with different p is shown by a different maker. Again we changed the values of α from 0.05 to 0.95 with an increment of 0.1.



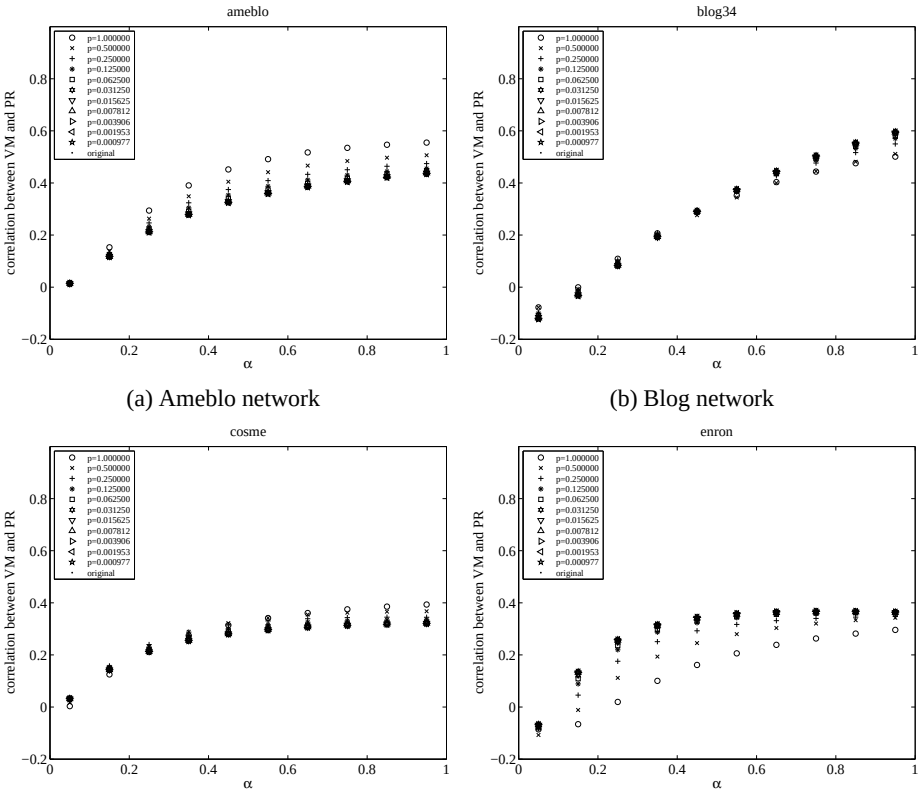
(a) Ameblo network (b) Blog network

(c) Cosme network (d) Enron network

Fig. 3: The correlation coefficient of VM between the original network and the rewiring network with p

From Fig.3, we can observe the following smilar characteristics of VM for all of the four networks. First, the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha, p))$ for any pair of α and p are relatively high. Second, in comparison to Fig.1, there exist almost no ranges for α where $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha, p))$ gives especially high values for all values of p. Third, $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha, p))$ monotonically decreases as p increases. Overall, this experimental results suggest that the expected influence degree is not much affected by the community structure although the effect is more for a network with less community structure.

Similarly to the above, let $\boldsymbol{y}(\beta, p)$ be the stationary vector calculated from the network rewired with probability p for PR with β. In order to examine the relation between VM and PR in terms of community structure, we calculated the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\beta, p))$ with respect to each pair of the uniform adoption probability α and the rewiring probability p by setting $\beta = \alpha$. In Fig.4, we plot $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\alpha, p))$ with respect to α, where each result with different p is shown by a different maker.



(a) Ameblo network (b) Blog network

(c) Cosme network (d) Enron network

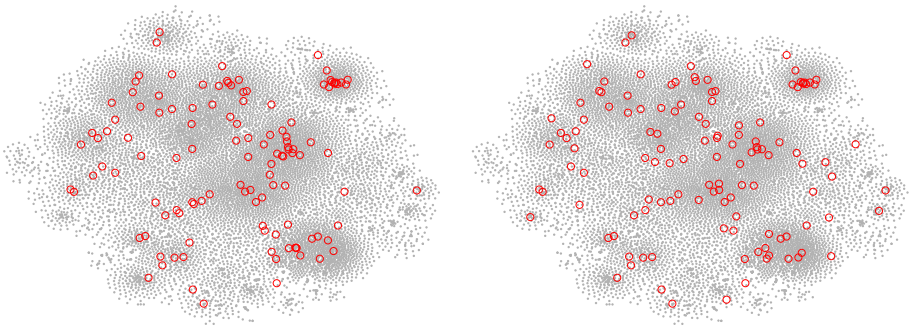
Fig. 4: The correlation coefficient between VM with α and PR with $\beta(=\alpha)$ and p

From Fig.4, we can see that for all of the four networks, each plotting result is very similar to the corresponding one appearing in Fig.2. Namely, the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\alpha, p))$ for any pair of α and p are relatively small. Further, this experimental results also suggest that the expected influence degree is not much affected by the community structure. As an interesting distinction, $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\alpha, p))$ is large when p is large for the Ameblo and Cosme networks, but a reverse tendency can be observed for the Blog and Enron networks. Clarifying this reason is left for our future work.

5.5 Visual analyses

We further analyzed the effects of the uniform adoption probability on the expected influence degree by visualizing the original networks. More specifically, we embeded the nodes in each network into a 2-dimensinal space by using the cross-entropy method [17], and plotted them as points. Then, we emphasized the highly influential nodes that have the expected influence degree within the top 1 % by using (red) circles. In the following experiments we only show the results using the Blog network as an example, but similar results were obtained for the other networks.

Fig.5 are the visualization results for two different values of α. Here we set α to 0.25 and 0.95 becauase they are considered to give the most and the least representative values for the expected influence degree as discussed in Section 5.3. From Fig.5, we can see that the highly influential nodes scatter around the entire the network for both α values. This partly explains the reason why the expected influence degree is not much affected by the community structure. This figure also shows that these two visualization results are close to each other.

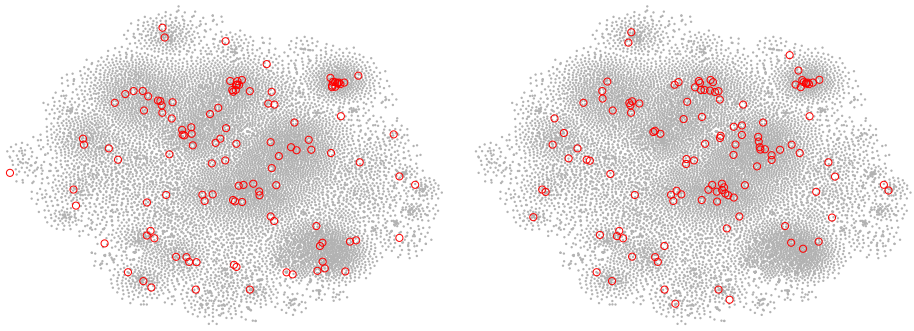


(a) Blog network at $\alpha = 0.25$ (b) Blog network at $\alpha = 0.95$

Fig. 5: Visualization of Networks (VM)

We also analyzed the resullts of PR to see if there is any difference between VM and PR. Fig.6 are the visualization results for PR, and the (red) circles are again the highly ranked nodes that have the top 1 % PageRank score. Here we set β to 0.25 and 0.95, the same as α. From Fig.6, we can also see that the highly ranked nodes scatter around

entire the network for both β values. Although we see that these nodes are different from the results of VM, but there is no clear difference bewteen the results of different β values.



(a) Blog network at $\beta = 0.25$ (b) Blog network at $\beta = 0.95$

Fig. 6: Visualization of Networks (PR)

6 Conclusion

We addressed in this paper the problem of estimating the influential nodes in a social network, and focused on a particular class of information diffusion model, a model for opinion propagation. The popular model for opinion propagation is the Voter model in which the main assumption is that people change their opinion based on their direct neighbors, i.e. via local interaction. We extended this model to include the fact that people's opinion is also affected by the overall opinion distribution of the whole society. The new model is called the Voter Model with uniform adoption (the extended VM). It assumes that the network is directional because the people to people relation is directional.

The uniform adoption implies the random opinion adoption of all nodes in the network. We came to notice that this mechanism is the same as the random surfer jump of the well known PageRank algorithm. This motivated us to investigate the relationship between the extended VM and PageRank. We mathematically derived the ranking vector of the extended VM and compared it with that of PageRank, and explored how the two models are related by a series of extensive experiments using four real world social networks. The both models assume a directed network and give different rankings because the adjacency matrix is asymmetric. However, if we assume an undirected network in which the adjacency matrix is symmetric, the both models become identical and should give the same ranking. We investigated the effects of the uniform adoption probability on node ranking and how the ranking of the extended VM and PageRank are correlated to each other with this probability. The results indicate that the correlation varies with the uniform adoptiopn probability. The correlation is very small when

the uniform adoption probability is small, but it becomes larger when both the uniform adoption and the random surfer jump probabilities become larger. However, the visualization rerults do not indicate the clear difference of the rankings between the different values of the uniform adoption probability. We also investigated how the different community structure affects the correlation, but did not see the strong effects. We found that the ranking becomes stable for the uniform adoption probability in the range of 0.15 and 0.35 and the self correlation within the extended Voter Model is high in this region. It is interesting to note that the reported recommended value for the random surfer jump of PageRank is 0.15, which is similar to our finding for the uniform adoption probability.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Agarwal, N., and Liu, H. (2008). Blogosphere: Research issues, tools, and applications. *SIGKDD Explorations* 10:18–31.
2. Brin, S. and Page, L. (1998) The anatomy of a large scale hypertextual Web search engine, In *Proceedings of the Seventh International World Wide Web Conference*, (pp. 107–117).
3. Crandall, D.; Cosley, D.; Huttenlocner, D.; Kleinberg, J.; and Suri, S. (2008). Feedback effects between similarity and sociai infiuence in online communities. In *Proceedings of KDD 2008*, (pp. 160–168).
4. Domingos, P., and Richardson, M. (2001). Mining the network value of customers. In *Proceedings of KDD 2001*, (pp. 57–66).
5. Even-Dar, E., and Shapria, A. (2007). A note on maximizing the spread of influence in social networks. In *Proceedings of WINE 2007*, (pp. 281–286).
6. Gruhl, D., Guha, R., Liben-Nowell, D., & Tomkins, A. (2004). Information diffusion through blogspace. *Proceedings of the 13th International World Wide Web Conference* (pp. 107–117).
7. Maximizing the spread of influence through a social network. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 137–146).
8. Kimura, M.; Saito, K.; Nakano, R.; and Motoda, H. (2010). Extracting influential nodes on a social network for information diffusion. *Data Mining and Knowledge Discovery* 20:70–97.
9. Klimt, B., and Yang, Y. (2004). The enron corpus: A new dataset for email classification research. In: Proceedings of the 2004 European Conference on Machine Learning (ECML'04). (pp. 217–226).
10. Langville, A. N. and Meyer, C. D. (2005). Deeper inside PageRank, *Internet Mathematics*, **1:3** 335–380.
11. Leskovec, J.; Adamic, L. A.; and Huberman, B. A. (2007). The dynamics of viral marketing. *ACM Transactions on the Web* 1:5.
12. Liggett, T. M. (1999). *Stochastic interacting systems: contact, voter, and exclusion processes*. New York: Springer.

13. Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, *45*, 167–256.
14. Richardson, M., and Domingos, P. (2002). Mining knowledge-sharing sites for viral marketing. In *Proceedings of KDD 2002*, (pp. 61–70).
15. Sood, V., and Redner, S. (2005). Voter model on heterogeneous graphs. *Physical Review Letters* 94:178701.
16. Wu, F., and Huberman, B. A. (2008). How public opinion forms. In *Proceedings of WINE 2008*, (pp. 334–341).
17. Yamada, T., Saito, K., & Ueda, N. (2003). Cross-entropy directed embedding of network data. *Proceedings of the 20th International Conference on Machine Learning* (pp. 832–839).
18. Yang, S.; Chen, W.; and Wang, Y. (2009). Efficient influence maximization in social networks. In *Proceedings of KDD 2009*, (pp. 199–208).

Efficient Estimation of Cumulative Influence for Multiple Activation Information Diffusion Model with Continuous Time Delay

Kazumi Saito[1], Masahiro Kimura[2], Kouzou Ohara[3], and Hiroshi Motoda[4]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[3] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We show that the node cumulative influence for a particular class of information diffusion model in which a node can be activated multiple times, i.e. Susceptible/Infective/ Susceptible (SIS) Model, can be very efficiently estimated in case of independent cascade (IC) framework with asynchronous time delay. The method exploits the property of continuous time delay within a stochastic framework and analytically derives the iterative formula to estimate cumulative influence without relying on awfully lengthy simulations. We show that it can accurately estimate the cumulative influence with much less computation time (about 2 to 6 orders of magnitude less) than the naive simulation using three real world social networks and thus it can be used to rank influential nodes quite effectively. Further, we show that the SIS model with a discrete time step, i.e. fixed synchronous time delay, gives adequate results only for a small time span.

1 Introduction

The proliferation of emails, blogs and social networking services (SNS) in the World Wide Web has accelerated the creation of large social networks [1–5]. Social networks naturally mediate the spread of various information. Innovation, topics and even malicious rumors can propagate in the form of so-called "word-of-mouth" communications. Thus, it is now understood that social networks provide rich sources of information that is useful to help understand the dynamics of our society, e.g. who are the best group of people to spread the desired information, how people respond to other people's opinion, what kind of topics propagate faster, how the public opinions are formed, how the way the information spreads differ from community to community, etc.

Several models have been proposed that simulate information diffusion through a network. The most widely-used model is the *independent cascade (IC)*. This is a fundamental probabilistic model of information diffusion [6, 7], which can be regarded as the so-called *susceptible/infective/recovered (SIR) model* for the spread of a disease [2]. This model has been used to solve such problems as the *influence maximization problem* which is to find a limited number of nodes that are influential for the spread of information [7, 8] and the *influence minimization problem* which is to suppress the spread of undesirable information by blocking a limited number of links [9]. Here, it is noted that the influence of a node is defined as the expected number of nodes that it can activate due to the stochastic nature of the information diffusion. The SIR model assumes that a node, once infected, never re-infected after it has been cured (recovered). Thus, the influence is normally defined as the expected number of recovered nodes at the end of the time span in consideration. The other class of model for the spread of a disease is the so-called *susceptible/infective/susceptible (SIS) model* [2], where a node, once infected, moves to a susceptible state and can be re-activated multiple times. A similar problem can be solved for this model, too [10, 11]. In these models, efficient methods of estimating the influence have been proposed based on bond percolation, strongly connected component decomposition, burnout and pruning [8, 11], but no analytical solutions have been found. Thus, efficiency remains that the computation time is 2 or 3 orders of magnitude faster than naive simulation.

The IC model above, whether it is used in SIR or SIS setting, cannot handle time-delays that are asynchronous and continuous for information propagation. Time step is incremented discretely and thus the node states are updated synchronously, which can be viewed that the time delay is fixed and synchronous. We call this "fixed time delay" for short. In reality, time flows continuously and thus information, too, propagates on this continuous time axis. For any node, information must be received at any time from any other nodes and must be allowed to propagate to yet other nodes at any other time with a possible delay, both in an asynchronous way. We call this "continuous time delay" for short. For example, the following scenario in case of SIS setting explains this need. Suppose a person A posted an article to a blog and a person B read it and responded a week later. Another person C posted an article on the same topic the next day A posted and B read it and responded the same day. B was activated twice, first by C and next by A although the time A was activated is earlier than C. Thus, for a realistic behavior analysis of information diffusion, we need to adopt a model that explicitly represents continuous asynchronous time delay. The continuous time delay SIR model was discussed in the machine learning problem setting in which the objective was to learn the parameters in the diffusion model from the observed time stamped node activation sequence data [3, 12]. In [12] it was shown that the parameters can be learned by maximizing the likelihood of the observed data being produced by the model. Note that there is no need to do simulation to obtain the influence degree in case of SIR setting because the final influence degree is equal to that of the model without time delay[1] since a node is not allowed to be re-activated multiple times.

In this paper, we address the problem of efficiently estimating the *cumulative influence* of a node in the network by adopting the information diffusion model that allows

[1] This is equivalent to fixed time delay in discrete time setting.

continuous time delay and multiple activation of the same node under the framework of independent cascade model, called CTSIS for short. Interestingly, although the model we considered in this paper is most complicated among the series of the models discussed above, it is possible to derive a formula analytically, under a simplified condition, that can iteratively estimate the *cumulative influence* of a node exploiting the property of continuous time delay within a stochastic framework. What makes the analysis easier is that in case of the continuous time there is only one single node that can be activated at a time, i,e, no multiple activations at different nodes at the same time, and no simultaneous activations of a node by its multiple active parents each of which has been activated at a different time in the past. Thus it does not make sense to define the node influence at a specific time and in light of SIS and continuous time delay we naturally define the influence to be an integral over a specified time span (*cumulative influence*), which is more meaningful in many practical settings.

We show that the proposed method (called *iterative method*) can accurately estimate the cumulative influence with much less computation time (about 2 to 6 orders of magnitude less) than empirical mean of the *naive simulation method* with a limited number of runs using three real world social networks with different sizes and connectivities. The method can be used to rank influential nodes quite effectively. We compare the proposed methods with two other methods, the SIS with fixed time delay and the one which is the extreme case of the propose method where the time span is set to be infinitely large (called *infinite iterative method*). We show that these are indeed less accurate and discuss under which conditions these work well, e.g. SIS with fixed time delay only works well for a small time span.

The paper is organized as follows. We revisit the information diffusion model, in particular SIS family, in section 2, and explain the proposed method of cumulative influence estimation in section 3. Then we report the experimental results in section 4, followed by discussion in section 5. We summarize our conclusion in section 6.

2 Information Diffusion Model

Let $G = (V, E)$ be a directed network, where V and E ($\subset V \times V$) stand for the sets of all the nodes and (directed) links, respectively. For any $v \in V$, let $\Gamma(v; G)$ denote the set of the child nodes (directed neighbors) of v, that is,

$$\Gamma(v; G) = \{w \in V;\ (v, w) \in E\}.$$

We consider information diffusion models on G in the susceptible/infected/susceptible (SIS) framework. In this context, infected nodes mean that they have just adopted the information, and we call these infected nodes *active* nodes.

2.1 Basic SIS Model

We first define the basic SIS model for information diffusion on G. In the model, the diffusion process unfolds in discrete time-steps $t \geq 0$, and it is assumed that the state of a node is either active or inactive. For every link $(u, v) \in E$, we specify a real value

$\kappa_{u,v}$ with $0 < \kappa_{u,v} < 1$ in advance. Here, $\kappa_{u,v}$ is referred to as the *diffusion parameter* through link (u, v). Given an initial active node v_0 and a time span T, the diffusion process proceeds in the following way. Suppose that node u becomes active at time-step t ($< T$). Then, node u attempts to activate every $v \in \Gamma(u; G)$, and succeeds with probability $\kappa_{u,v}$. If node u succeeds, then node v will become active at time-step $t + 1$. Thus, as mentioned in 1, we can view this as synchronous fixed time delay[2]. If multiple active nodes attempt to activate node v in time-step t, then their activation attempts are sequenced in an arbitrary order. On the other hand, node u will become inactive at time-step $t + 1$ unless it is activated by an active node in time-step t. The process terminates if the current time-step reaches the final time T.

2.2 Continuous-time SIS model

Next, we extend the basic SIS model so as to allow continuous-time delays, and refer to the extended model as the *continuous-time SIS (CTSIS) model*[3]. This model can be interpreted as *susceptible/exposed/infective/susceptible (SEIS) model* in that a node does not become active (infected) instantly when activated, but wait for a while (exposed) before it gets activated (infected).

In the CTSIS model on G, for each link $(u, v) \in E$, we specify real values $r_{u,v}$ and $\kappa_{u,v}$ with $r_{u,v} > 0$ and $0 < \kappa_{u,v} < 1$ in advance. We refer to $r_{u,v}$ and $\kappa_{u,v}$ as the *time-delay parameter* and the *diffusion parameter* through link (u, v), respectively.

Let T be the time span. The diffusion process unfolds in continuous-time t, and proceeds from a given initial active node v_0 in the following way. Suppose that a node u becomes active at time t ($< T$). Then a delay-time δ is chosen for u's every child node $v \in \Gamma(u; G)$ from the exponential distribution with parameter $r_{u,v}$. If $t + \delta \leq T$, v is activated by u with success probability $\kappa_{u,v}$ at $t + \delta \leq T$. Under the continuous time framework, there is no possibility that multiple parent nodes of v simultaneously activate v exactly at the same time $t + \delta$. On the other hand, node u will become or remain inactive at time t' ($> t$) unless it is activated from other active nodes. The process terminates if the current time reaches the final time T.

2.3 Influence Function

Let T be the time span for the CTSIS model on G. We consider a time-interval $[T_0, T_1]$ with $0 \leq T_0 < T_1 \leq T$. For any node $v \in V$, let $S(v; T_0, T_1)$ denote the total number of active nodes within time-interval $[T_0, T_1]$ for the probabilistic diffusion process from an initial active node v under the CTSIS model. Note that $S(v; T_0, T_1)$ is a random variable. Let $\sigma(v; T_0, T_1)$ denote the expected value of $S(v; T_0, T_1)$. We call $\sigma(v; T_0, T_1)$ the *cumulative influence degree* of node v within time-interval $[T_0, T_1]$. Note that σ is a function defined on V. We call the function $\sigma(\cdot; T_0, T_1) : V \to \mathbf{R}$ the *cumulative influence function* for the CTSIS model within time-interval $[T_0, T_1]$ on network G.

[2] This may well be called as "no time delay" because time delay is not explicitly represented in the formulation.

[3] Note that the information propagates at a certain time point, but its delay can be continuous.

It is important to estimate the cumulative influence function $\sigma(\cdot; T_0, T_1)$ efficiently. In theory we can simply estimate it by simulating the CTSIS model in the following way. First, a sufficiently large positive integer M is specified. For each $v \in V$, the diffusion process of the CTSIS model is simulated from initial active node v, and the total number of active nodes within time-interval $[T_0, T_1]$, $S(v; T_0, T_1)$, is calculated. Then, $\sigma(v; T_0, T_1)$ is estimated as the empirical mean of $S(v; T_0, T_1)$ that are obtained from M such simulations. We refer to this estimation method as the *naive simulation method*. However, as shown in the experiments, this is extremely inefficient, and cannot be practical (out of question). In this paper, we deal with the case "$T_0 = 0$, $T_1 = T$" for simplicity, and we denote $\sigma(v; 0, T)$ by $\sigma(v; T)$.

3 Estimation Methods

For a given directed graph $G = (V, E)$, we identify each node with a unique integer from 1 to $|V|$. Then we can define the adjacency matrix $\boldsymbol{A} \in \{0, 1\}^{|V|\times|V|}$ by setting $a_{u,v} = 1$ if $(u, v) \in E$; otherwise $a_{u,v} = 0$. We also define the probability matrix $\boldsymbol{P} \in [0, 1)^{|V|\times|V|}$ by replacing each element $a_{u,v}$ to the corresponding diffusion probability $\kappa_{u,v}$ if $(u, v) \in E$. Let $\boldsymbol{f}_v \in \{0, 1\}^{|V|}$ be a vector whose v-th element is 1 and other elements are 0, and $\mathbf{1} \in \{1\}^{|V|}$ be a vector whose elements are all 1.

3.1 Infinite Iterative Method

We can calculate the number of nodes that are reachable with J-steps starting from a node v by $\boldsymbol{f}_v^T \boldsymbol{A}^J \mathbf{1}$. Thus, when considering the diffusion probabilities, we can calculate the vector of the expected number of reachable nodes starting from each node within J steps by $\boldsymbol{P}\mathbf{1} + \cdots + \boldsymbol{P}^J\mathbf{1}$. Therefor, in case that the time-interval is $[0, \infty]$, according to the definition of the CTSIS model, we obtain the cumulative influence degree $\boldsymbol{\sigma}_\infty$ as follows:

$$\boldsymbol{\sigma}_\infty = \sum_{J=1}^{\infty} \boldsymbol{P}^J \mathbf{1}, \tag{1}$$

Note that the vector $\boldsymbol{\sigma}_\infty$ consists of values of the cumulative influence functions, i.e., $\sigma(\cdot; \infty)$. We refer to this estimation method as the *infinite iterative method*.

However, there exist some intrinsic limitations to the simple iterative method, i.e., we cannot specify arbitrary time-interval $[T_0, T_1]$ and diffusion probabilities for this method. As for the diffusion probabilities, when the largest eigenvalue of the probability matrix $\boldsymbol{P}$ is less than 1, we can guarantee to obtain finite value of $\boldsymbol{\sigma}_\infty$. In a simple case that the diffusion parameters are uniform for any link, i.e., $\kappa_{u,v} = \kappa$ for any $(u, v) \in E$, since the probability matrix $\boldsymbol{P}$ is equivalent to $\kappa\boldsymbol{A}$, the diffusion parameter κ must be less than the reciprocal of the the largest eigenvalue of the adjacency matrix $\boldsymbol{A}$. Incidentally, the calculation formula for this simple case is quite similar to that of Bonacich's centrality [13] and identical to that of Katz's measure [14].

3.2 Proposed Method

We want to estimate the cumulative influence degree within time-interval $[T_0, T_1]$ for arbitrary diffusion probabilities. To this end, we introduce the probability $R(J; T_0, T_1)$ that diffusion takes J-steps within this time-interval according to the CTSIS model. Here, in order to simplify our derivation, we focus on the simplest case that the time-delay parameters are uniform for any link, i.e., $r_{u,v} = r$ for any $(u, v) \in E$, although our approach can be naturally extended to more complex settings. In a special case where $T_0 = 0$ and $T_1 = T$, we denote this probability by $R(J; T)$. Here we note that $R(S; T_0, T_1) = R(S; T_1) - R(S; T_0)$. Thus we focus on calculation of $R(J; T)$.

Let δ_j be a random variable of a time-delay for the j-th step ($1 \leq j \leq J$). In order to meet the condition that the diffusion takes J-steps within time-interval $[0, T]$, the total sum of the time-delays must be less than T, i.e., $0 \leq \delta_1 + \cdots + \delta_S \leq T$. In case of $J = 1$, we can easily obtain the following formula.

$$R(1; T) = \int_0^T r \exp(-r\delta_1) d\delta_1 = 1 - \exp(-rT). \tag{2}$$

In case of $J \geq 2$, due to the independence of time-delay trials, we can calculate the probability $R(J; T)$ as follows:

$$R(J; T) = \int_0^T \int_0^{T-\delta_1} \cdots \int_0^{T-(\delta_1+\cdots+\delta_{J-1})} \prod_{j=1}^{J} r \exp(-r\delta_j) d\delta_1 \cdots d\delta_J \tag{3}$$

Here by noting the following two formulas,

$$\int_0^{T-(\delta_1+\cdots+\delta_{J-1})} r \exp(-r\delta_J) d\delta_J = 1 - \exp(-rT) \prod_{j=1}^{J-1} \exp(r\delta_j),$$

$$\int_0^T \cdots \int_0^{T-(\delta_1+\cdots+\delta_{J-2})} r^{J-1} \exp(-rT) d\delta_1 \cdots d\delta_{J-1} = \exp(-rT) \frac{(rT)^{J-1}}{(J-1)!},$$

we can calculate Eq. 3 as follows:

$$R(J; T) = R(J-1; T) - \exp(-rT) \frac{(rT)^{J-1}}{(J-1)!} \tag{4}$$

Therefore, from Eqs. 2 and 4, we can derive the following explicit formula:

$$R(J; T) = 1 - \exp(-rT) \sum_{j=1}^{J} \frac{(rT)^{j-1}}{(j-1)!}. \tag{5}$$

Here, we can easily see that $R(J; T)$ is a monotonic decreasing function approaching to zero as J increases.

Now, by combining Eqs. 1 and 5, we can derive a new method for estimating the cumulative influence degree within time-interval $[T_0, T_1]$ for arbitrary diffusion probabilities. We can formulate the key formula as follows:

$$\boldsymbol{\sigma}_{[T_0,T_1]} = \sum_{J=1}^{\infty} R(J; T_0, T_1) \boldsymbol{P}^J \mathbf{1}. \tag{6}$$

Below we can summarize the algorithm of the proposed method.

1. Set each element of $\boldsymbol{\sigma}_{[T_0,T_1]}$ to 0, and set $J \leftarrow 1$ and $\boldsymbol{x} \leftarrow \mathbf{1}$.
2. Calculate $\boldsymbol{x} \leftarrow \boldsymbol{P}\boldsymbol{x}$ and if $R(J; T_0, T_1)\|\boldsymbol{x}\| < \eta$, then output $\boldsymbol{\sigma}_{[T_0,T_1]}$ and terminate.
3. Set $\boldsymbol{\sigma}_{[T_0,T_1]} \leftarrow \boldsymbol{\sigma}_{[T_0,T_1]} + R(J; T_0, T_1)\boldsymbol{x}$ and $J \leftarrow J + 1$ and return to 2.

In this algorithm, $\boldsymbol{x} \in \mathbb{R}^{|V|}$ is a vector to calculate the expected number of the J-step reachable nodes, and η is a parameter for the termination condition. In our experiments, η is set to a sufficiently small number, i.e., 10^{-12}.

4 Experiments

We first evaluate the performance (accuracy) of the proposed method (*iterative method*) by comparing with the *naive simulation method* with different number of runs to estimate the empirical mean using three large real social networks. We then compare the *iterative method* with two other methods, the *infinite iterative method* and the *SIS with fixed time delay method* in terms of the estimated *cumulative influence degree* for the CTSIS model using the same networks. Finally we compare the efficiency (computation time) of the *iterative method* with the *naive simulation method*. In all the experiments, we consider the simplest case where the both diffusion and time-delay parameters of the CTSIS model are uniform for any link.

4.1 Datasets

We employed three datasets of large real networks. These are all bidirectionally connected networks. The first one is a network of people that was derived from the "list of people" within Japanese Wikipedia, also used in [15], and has $9,481$ nodes and $245,044$ directed links (the Wikipedia network). The second one is a network derived from the Enron Email Dataset [16] by extracting the senders and the recipients and linking those that had bidirectional communications, and has $4,254$ nodes and $44,314$ directed links (the Enron network). The third one is a Coauthorship network used in [17] and has $12,357$ nodes and $38,896$ directed links (the coauthorship network).

4.2 Accuracy Evaluation

We evaluated the accuracy of the proposed method by comparing it with the *naive simulation method* mentioned in section 2.3. We speculate that the *cumulative influence degree* estimated by taking the empirical mean of the results of the *naive simulation method* converges asymptotically to the true value as the number of simulations M increases. Thus, we first examined how the difference of the estimated cumulative influence degree between the *iterative method* and the *naive simulation method* changes as M changes for the three networks.

The difference was evaluated by

$$\epsilon_M = \sum_{v \in V} |\sigma(v; T) - s_M(v; T)|/|V|, \tag{7}$$

where $\sigma(v;T)$ and $s_M(v;T)$ are the *cumulative influence degree* of node v estimated by the *iterative method* and the *naive simulation method*, respectively. We used $T = 10^4$ and varied M from 100, 1,000, and 10,000.

In these experiments we determined the values for the diffusion and time-delay parameters as follows. As noted in 3.1, it is required that the diffusion parameter κ must be less than $eig(\boldsymbol{A})^{-1}$, the reciprocal of the largest eigenvalue of the adjacency matrix $\boldsymbol{A}$ of the network for the *infinite iterative method* to obtain a finite value of $\boldsymbol{\sigma}_\infty$. The values of $eig(\boldsymbol{A})^{-1}$ for the Wikipedia, Enron, and Coauthorship networks were 0.00674, 0.0205, and 0.105, respectively. Thus, we adopted 0.0067, 0.02, and 0.1 as the values of κ for these networks, respectively. These are the largest values that the *infinite iterative method* can take. We set $r = 1$ for the time-delay parameter. This is equivalent to setting the average time delay to be a unit time which is consistent to the discrete time step of the *SIS with fixed time delay* method.

Table 1 summarizes the results, from which we can see that the estimation difference decreases as M increases and it becomes reasonably small at $M = 10,000$ for all the three networks. We are able to verify our speculation and conclude that the proposed *iterative method* can indeed estimate the *cumulative influence* accurately.

Table 1: Estimation difference between the *iterative method* (proposed) and the *naive simulation method*

network	M		
	100	1,000	10,000
Wikipedia	0.196	0.062	0.020
Enron	0.552	0.190	0.062
Coauthorship	0.298	0.096	0.031

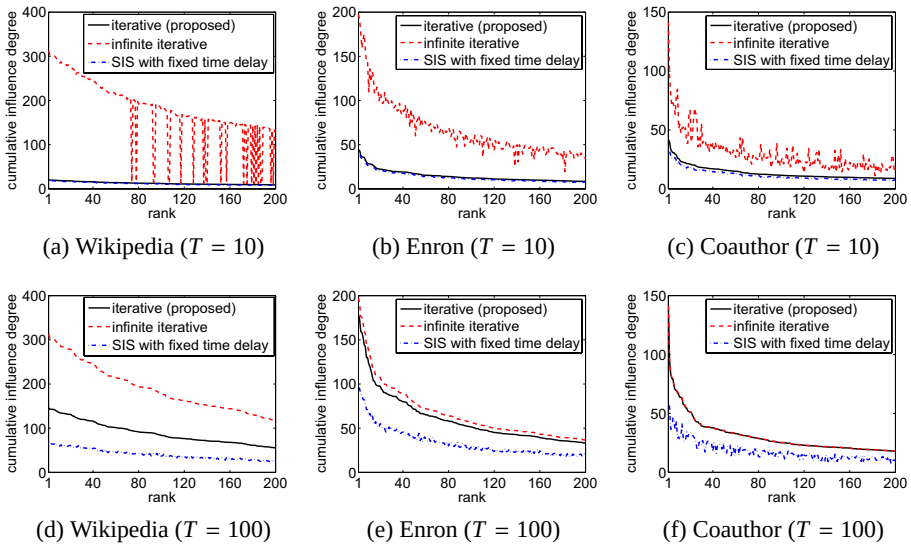
4.3 Cumulative Influence Degree Comparison

Next, we investigated how well the other approaches can approximate the *cumulative influence degree*. We compared two approaches. One is the *infinite iterative method* described in 3.1. The other is the *SIS with fixed time delay* method [11][5]. The *SIS with fixed time delay* method uses bond percolation on the layered graph which is constructed from the original social network with each layer added on top as the time proceeds[10] and much more efficiently estimates the *cumulative influence degree* than the *naive simulation method*. We used the same M (= 10,000) from the result in 4.2. For each network, we investigated two cases, one with a short time span $T = 10$ and the other with a long time span $T = 100$. Note that we set r=1 and thus, the average time delay $\overline{\delta} = 1$. We selected the top 200 most influential nodes that the *iterative method* identified and compared their *cumulative influence degree* with the values that the other two methods estimated for the same 200 nodes.

Figure 1 illustrates the results of comparison. We can see that the *infinite iterative method* estimate the *cumulative influence degree* fairly well for a long time span $T = 100$ except for the Wikipedia network, but it tends to overestimate it for a short time span $T = 10$. In contrast, the *SIS with fixed time delay method* tends to underestimate

[4] We had to set the value to be small so that the naive simulation returns the result within a day.

[5] Note that in [11] the influence degree was defined to be the expected number of active nodes at the end of observation time T, but here the algorithm in [11] is modified to calculated the *cumulative influence degree*.



(a) Wikipedia ($T = 10$) (b) Enron ($T = 10$) (c) Coauthor ($T = 10$)

(d) Wikipedia ($T = 100$) (e) Enron ($T = 100$) (f) Coauthor ($T = 100$)

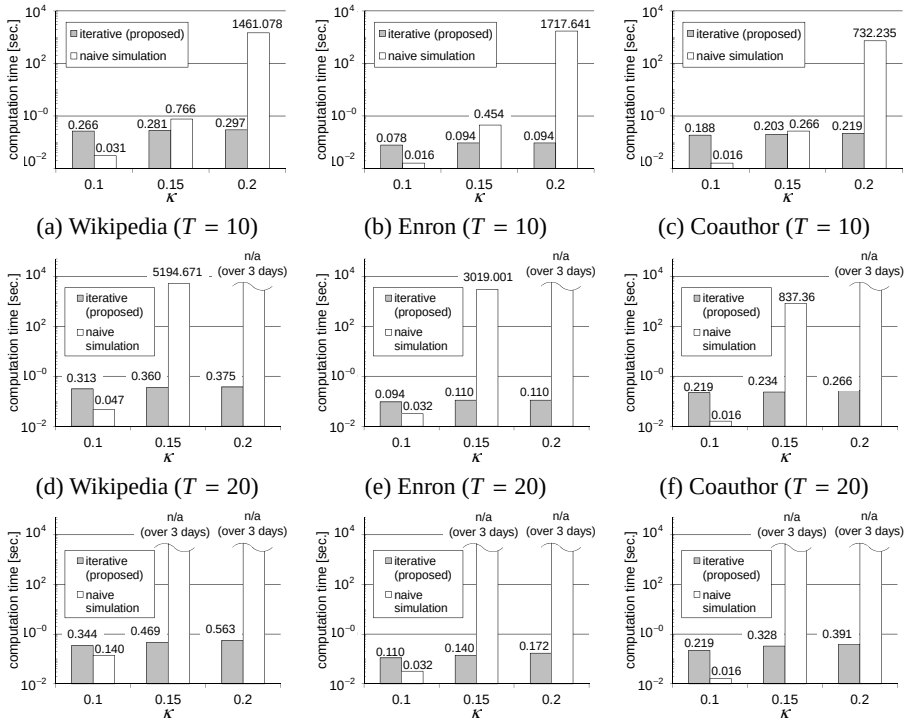
Fig. 1: Comparison in cumulative influence degrees of top 200 influential nodes

the *cumulative influence degree* for a large time span $T = 100$ but it does well for a short time span $T = 10$. These results show that these two methods cannot correctly estimate the *cumulative influence degree* for an arbitrary time span.

It is noted that there are many bumps in the graphs for the cases where the estimation of the other two methods is very poor, i.e. $T = 10$ for the *infinite iterative method* and $T = 100$ for the *SIS with fixed time delay method*. This implies that the ranking results by these methods are different from the true ranking by the *iterative method*. The curves becomes smoother when the estimation becomes better.

4.4 Efficiency Evaluation

We see in 4.3 that both *infinite iterative method* and *SIS with fixed time delay method* do not accurately estimate the *cumulative influence degree*, and we compare the computation time of the *iterative method* with the *naive simulation method* for $M = 1$. The results are shown in Fig. 2 for three values of the time span T= 10, 20, 100 and for each of the three networks. Three values are chosen for κ. The minimum values are the same as the ones used in 4.2 and 4.3, and the other values are obtained by multiplying 1.5 in sequence. The *iterative method* returns the values in less than 0.5 sec. for all cases and very insensitive to the parameter values. The *native simulation method* is only efficient when the κ is very small and requires exponentially increasing time as κ increase. In deed it did not return the values within 3 days in many cases. Considering that this is for a single simulation, use of the *naive simulation method* is not practical and out of question.



(a) Wikipedia ($T = 10$) (b) Enron ($T = 10$) (c) Coauthor ($T = 10$)

(d) Wikipedia ($T = 20$) (e) Enron ($T = 20$) (f) Coauthor ($T = 20$)

(g) Wikipedia ($T = 100$) (h) Enron ($T = 100$) (i) Coauthor ($T = 100$)

Fig. 2: Comparison in computation time

5 Discussions

We mentioned in 3.1 that the *cumulative influence degree* derived by the *infinite iterative method* is similar to the centrality proposed by Bonacich [13] and identical to the Katz' measure [14]. In [13] the standard centrality e_u of node u is defined by

$$\lambda e_u = \sum_{v \in V} a_{u,v} e_v, \tag{8}$$

where λ is a constant introduced to ensure a non-zero solution, and $\boldsymbol{A}$ is the adjacency matrix ($a_{u,v}$ is its element) as before. Bonacich generalized Eq. 8 by introducing the strength of relationship β, which is equivalent to κ in this paper, and derived the generalized centrality $c_u(\alpha,\beta)$ as

$$c_u(\alpha,\beta) = \sum_{v \in V} (\alpha + \beta c_v(\alpha,\beta)) a_{u,v}, \tag{9}$$

where α is a normalization constant. It is easily shown that $c_i(\alpha,\beta)$ is written in a matrix notation as

$$\boldsymbol{c}(\alpha,\beta) = \alpha \sum_{J=0}^{\infty} \beta^J \boldsymbol{A}^{J+1} \mathbf{1} = \alpha(\boldsymbol{A}\mathbf{1} + \beta \boldsymbol{A}^2 \mathbf{1} + \beta^2 \boldsymbol{A}^3 \mathbf{1} + \cdots). \tag{10}$$

Comparing Eq. 1 with Eq. 10, we note that they are the same except that the generalized centrality assumes that the strength of relationship with the directed connected nodes is 1. Further, we note that the following equality holds.

$$\boldsymbol{\sigma}_{\infty} = \frac{\beta}{\alpha}\boldsymbol{c}(\alpha,\beta), \tag{11}$$

which is exactly the same as Katz's measure. Thus, the *cumulative influence degree* $\boldsymbol{\sigma}_{\infty}$ defined by the *infinite iterative method* is interpreted as a centrality measure.

We showed in 4.3 that the *infinite iterative method* well approximates the *cumulative influence degree* when the time span is large. This is evident because the *infinite iterative method* assumes an infinite time span. In the extreme limit of $T = \infty$, the *iterative method* converges to the infinite iterative method. How large T should be in order for it to be large depends on the delay time parameter r. When r gets smaller, a smaller T can be called large, e.g. $T = 10$ is large when $r = 0.1$. Similar argument can be made for the *SIS with fixed time delay method*. The *SIS with fixed time delay method* advances the time in a discrete step. Thus, it happens that multiple parents attempt to activate the same node simultaneously at the same time. If this happens, the activation count is only incremented by one. When the time span T is small, the diffusion propagation does not go far and there is not much chance that this simultaneous activation happens. This is why the *SIS with fixed time delay method* gives good results for a small time span T. However, how good the *SIS with fixed time delay method* approximates the *cumulative influence degree* depends on how close the time step is to the average delay-time $\overline{\delta}$. It overestimates the true *cumulative influence degree* for $T = 10$ when $r = 0.1$ and underestimate it when $r = 10$. We confirmed this by additional experiments but due to the space limit we do not show the figures.

6 Conclusion

In this paper we addressed the problem of efficiently estimating the *cumulative influence degree* of a node in social networks when the information diffusion follows the Susceptible/Infective/Susceptible (SIS) model with asynchronous continuous time delay based on the independent cascade framework. It is possible to analytically derive a formula by which to iteratively calculate the *cumulative influence degree* to a desired accuracy. The simplified version which corresponds to assuming an infinitely large time span is closely related to the generalized centrality measure. We showed by applying the method to three large real world social networks that the method can accurately estimate the *cumulative influence degree* with 2 to 6 orders of magnitude less computation time than the *naive simulation method*. Thus, it can be used to rank the influential nodes very efficiently. We also compared the proposed *iterative method* to the *SIS with fixed time delay model* and the *infinite iterative method* and confirmed that they generally produce poor estimates and only give good results when a specific condition holds for each.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-08-4027, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Physical Review E **66** (2002) 035101
2. Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45** (2003) 167–256
3. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. SIGKDD Explorations **6** (2004) 43–52
4. Domingos, P.: Mining social networks for viral marketing. IEEE Intelligent Systems **20** (2005) 80–82
5. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06). (2006) 228–237
6. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters **12** (2001) 211–223
7. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003). (2003) 137–146
8. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07). (2007) 1371–1376
9. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data **3** (2009) 9:1–9:23
10. Kimura, M., Saito, K., Motoda, H.: Efficient estimation of influence functions fot sis model on social networks. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09). (2009) 2046–2051
11. Saito, K., Kimura, M., Motoda, H.: Discovering influential nodes for SIS models in social networks. In: Proc. of the Twelfth International Conference of Discovery Science (DS2009), LNAI 5808. (2009) 302–316
12. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Learning continuous-time information diffusion model for social behavioral data analysis. In: Proc. of the First Asian Conference on Machine Learning, LNAI 5828. (2009) 322–337
13. Bonacichi, P.: Power and centrality: A family of measures. American Journal of Sociology **92** (1987) 1170–1182
14. Katz, L.: A new status index derived from sociometric analysis. Sociometry **18** (1953) 39–43
15. Kimura, M., Saito, K., Motoda, H.: Minimizing the spread of contamination by blocking links in a network. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08). (2008) 1175–1180
16. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: Proceedings of the 2004 European Conference on Machine Learning (ECML'04). (2004) 217–226
17. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435** (2005) 814–818

Learning to Predict Opinion Share in Social Networks

Masahiro Kimura
Department of Electronics and Informatics
Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp

Kazumi Saito
School of Administration and Informatics
University of Shizuoka
Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp

Kouzou Ohara
Department of Integrated Information Technology
Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp

Hiroshi Motoda
Institute of Scientific and Industrial Research
Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract

We address the problem of predicting the expected opinion share over a social network at a target time from the opinion diffusion data under the value-weighted voter model with multiple opinions. The value update algorithm ensures that it converges to a correct solution and the share prediction results outperform a simple linear extrapolation approximation when the available data is limited. We further show in an extreme case of complete network that the opinion with the highest value eventually takes over, and the expected share prediction problem with uniform opinion value is not well-defined and any opinion can win.

Introduction

Blogosphere and sites such as for social networking, knowledge-sharing and media-sharing in the World Wide Web have enabled to form various kinds of large social networks, through which behaviors, ideas and opinions can spread. Thus, substantial attention has been directed to investigating the spread of influence in these networks (Leskovec, Adamic, and Huberman 2007; Crandall et al. 2008; Wu and Huberman 2008).

The representative problem is the *influence maximization problem*, that is, the problem of finding a limited number of influential nodes that are effective for the spread of information through the network and new algorithmic approaches have been proposed under different model assumptions, e.g., descriptive probabilistic interaction models (Domingos and Richardson 2001; Richardson and Domingos 2002), and basic diffusion models such as *independent cascade (IC) model* and the *linear threshold (LT) model* (Kempe, Kleinberg, and Tardos 2003; Kimura et al. 2010; Chen, Wang, and Yang 2009). This problem has good applications in sociology and "viral marketing" (Agarwal and Liu 2008). The models used above allow a node in the network to take only one of the two states, i.e., either active or inactive, because the focus is on *influence*.

However, application such as an on-line competitive service in which a user can choose one from multiple choices/decisions requires a model that handles multiple states. Further, it is important to consider the value of each choice, e.g., quality, brand, authority, etc. because this impacts other's choice. We formulate this problem as a value-weighted K opinion diffusion problem and provides a way to accurately predict the expected share of the opinions at a future target time T (before an consensus is reached) from a limited amount of observed data.

A good model for opinion dynamics would be a voter model. It is one of the most basic stochastic process model, and has the same key property with the *linear threshold (LT) model* that a node decision is influenced by its neighbor's decision, i.e., a person changes its opinion by the opinions of its neighbors. In the basic voter model which is defined on an undirected network, each node initially holds one of K opinions, and adopts the opinion of a randomly chosen neighbor at each subsequent discrete time-step.

There has been a variety of work on the voter model. Dynamical properties of the basic model, including how the degree distribution and the network size affect the mean time to reach consensus, have been extensively studied (Liggett 1999; Sood and Redner 2005) from mathematical point of view. Several variants of the voter model are also investigated (Castellano, Munoz, and Pastor-Satorras 2009; Yang et al. 2009) and non equilibrium phase transition is analyzed from physics point of view. Yet another line of work extends the voter model and combine it with a network evolution model (Holme and Newman 2006; Crandall et al. 2008). The major interests there are different from what this paper intends to address, i.e., share prediction at a specific time T with opinion values considered.

Even-Dar and Shapira (2007) investigated the influence maximization problem (maximizing the spread of the opinion that supports a new technology) under the basic *voter model* with two ($K = 2$) opinions (one in favor of the new technology and the other against it) at a given target time T. They showed that the most natural heuristic solution, which picks the nodes in the network with the highest degree, is indeed the optimal solution, under the condition that all nodes have the same cost. This work is close to ours in that it measures the influence at a specific time T but is different in all others (no share prediction, no value considered, $K = 2$, no asynchronous update and no learning).

To the best of our knowledge, there has been no study that tried to predict the future opinion shares from the limited observed data in machine learning framework for the problem

of modeling the diffusion of several competitive opinions in a social network based on the voter model with opinion values considered. We learn the values of opinions from the limited amount of observed opinion diffusion data (i.e., data from 0 to T_0) and use the estimated values to predict the future (i.e., share at T ($> T_0$)). We show that the proposed approach works very satisfactorily using two real world social networks, and further a simple theoretical analysis reveals that it is indeed crucial to consider the opinion values and accurately estimate them for share prediction.

Our contribution is that 1) we proposed an algorithm that ensures the global optimal solution for the opinion value estimation from the observed opinion diffusion data, 2) we showed that the estimated model can accurately predict the future expected opinion share and outperforms the simple linear extrapolation prediction, and that, in the extreme case where all the nodes are connected to each other (i.e., complete network), 3) the opinion share prediction problem is not well-defined without introduction of opinion values and any opinion can prevail, and 4) the consensus is reached at which the opinion with the highest value wins and all the others die.

Opinion Dynamics

We consider the diffusion of opinions in a social network represented by an undirected (bidirectional) graph $G = (V, E)$ with self-loops. Here, V and E ($\subset V \times V$) are the sets of all the nodes and links in the network, respectively. For a node $v \in V$, let $\Gamma(v)$ denote the set of neighbors of v in G, that is, $\Gamma(v) = \{u \in V;\ (u, v) \in E\}$. Note that $v \in \Gamma(v)$.

Voter Model

According to the work (Even-Dar and Shapria 2007), we recall the definition of the basic voter model with two opinions on network G. In the voter model, each node of G is endowed with two states; opinions 1 and 2. The opinions are initially assigned to all the nodes in G, and the evolution process unfolds in discrete time-steps $t = 1, 2, 3, \cdots$ as follows: At each time-step t, each node v picks a random neighbor u and adopts the opinion that u holds at time-step $t-1$.

More formally, let $f_t : V \to \{1, 2\}$ denote the *opinion distribution* at time-step t, where $f_t(v)$ stands for the opinion of node v at time-step t. Then, $f_0 : V \to \{1, 2\}$ is the initial opinion distribution, and $f_t : V \to \{1, 2\}$ is inductively defined as follows: For any $v \in V$,

$$\begin{cases} f_t(v) = 1, & \text{with probability } \frac{n_1(t-1,v)}{n_1(t-1,v) + n_2(t-1,v)}, \\ f_t(v) = 2, & \text{with probability } \frac{n_2(t-1,v)}{n_1(t-1,v) + n_2(t-1,v)}, \end{cases}$$

where $n_k(t, v)$ is the number of v's neighbors that hold opinion k at time-step t for $k = 1, 2$.

Value-weighted Voter Model

We extend the original voter model for our purpose. In our model, the total number of opinions is set to K (≥ 2), and each node of G is endowed with $(K + 1)$ states; opinions $1, \cdots, K$, and *neutral* (i.e., no-opinion state). We consider that a node is *active* when it holds an opinion k, and a node is *inactive* when it does not have any opinion (i.e., its state is neutral). We assume that nodes never switch their states from active to inactive. In order to discuss the competitive diffusion of K opinions, we introduce the *value parameter* w_k (> 0) for each opinion k. In the same way as the original voter model, let $f_t : V \to \{0, 1, 2, \cdots, K\}$ denote the opinion distribution at time t, where opinion 0 denotes the neutral state. We also denote by $n_k(t, v)$ the number of v's neighbors that hold opinion k at time t for $k = 1, 2, \cdots, K$, i.e.,

$$n_k(t, v) = |\{u \in \Gamma(v);\ f_t(u) = k\}|.$$

We start the evolution process from an initial state in which each opinion is assigned to only one node and all other nodes are in the neutral state. Given a target time T, the evolution process unfolds in the following way. In general, each node v considers changing its opinion based on the current opinions of its neighbors at its $(j-1)$th update-time $t_{j-1}(v)$, and actually changes its opinion at the jth update-time $t_j(v)$, where $t_{j-1}(v) < t_j(v) \leq T$, $j = 1, 2, 3, \cdots$, and $t_0(v) = 0$. It is noted that since node v is included in its neighbors by definition, its own opinion is also reflected. The jth update-time $t_j(v)$ is decided at time $t_{j-1}(v)$ according to the exponential distribution of parameter λ (we simply use $\lambda = 1$ for any $v \in V$ in our experiments)[1]. Then, node v changes its opinion at time $t_j(v)$ as follows: If node v has at least one active neighbor at time $t_{j-1}(v)$,

$$f_{t_j(v)}(v) = k, \quad \text{with probability } \frac{w_k\, n_k(t_{j-1}(v), v)}{\sum_{k'=1}^{K} w_{k'}\, n_{k'}(t_{j-1}(v), v)}$$

for $k = 1, \cdots, K$, otherwise,

$$f_{t_j(v)}(v) = 0, \quad \text{with probability } 1.$$

Note here that $f_t(v) = f_{t_{j-1}(v)}(v)$ for $t_{j-1}(v) \leq t < t_j(v)$. If the next update-time $t_j(v)$ pasts T, that is, $t_j(v) > T$, then the opinion evolution of v is over. The evolution process terminates when the opinion evolution of every node in G is over.

Opinion Share Prediction

Based on our opinion dynamics model, we investigate the problem of predicting how large a share each opinion will have at a future target time T when the opinion diffusion is observed from $t_0 (= 0)$ to T_0 ($< T$). Let $\mathcal{D}_{T_0}$ be the observed opinion diffusion data in time-interval $[0, T_0]$, that is,

$$\mathcal{D}_{T_0} = \{(v, t, f_t(v));\ v \in V,\ t = 0, t_1(v), \cdots, t_{J_v}(v)\}.$$

Note that $t_{J_v}(v) \leq T_0$ for every $v \in V$. We define the *population* $h_k(t)$ of opinion k at time t by

$$h_k(t) = |\{v \in V;\ f_t(v) = k\}|$$

for $k = 1, 2, \cdots, K$.

[1]Note that this is equivalent to picking a node randomly and updating its opinion in turn $|V|$ times.

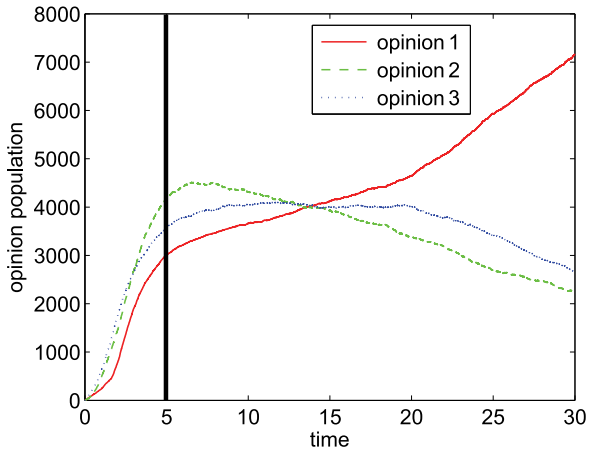


Figure 1: An example of opinion population curves in the blog network for $K = 3$.

Figure 1 shows an example of opinion population curves $h_1(t)$, $h_2(t)$, $h_3(t)$ for $K = 3$ in the blog network (see the section of "Experimental Evaluation" below), where $w_1 = 1.5$, $w_2 = 1.0$, $w_3 = 1.1$. Here, if we set $T_0 = 5$ and $T = 30$, we are able to observe $\mathcal{D}_5$ and thus $\{h_k(t);\ 0 \leq t \leq 5\}$ for $k = 1, 2, 3$ and the problem is to predict $h_1(30)$, $h_2(30)$, $h_3(30)$. Note that although the opinion dynamics is stochastic, we found that the variance of the value of $h_k(30)$ $(k = 1, 2, 3)$ is relatively small for $T_0 = 5$. We can easily see from Figure 1 that the naive time-series analysis method does not work well for this prediction problem. Thus, it is crucial to accurately estimate the values of w_1, w_2 and w_3 from the observed opinion diffusion data.

We define the *share* $g_k(t)$ of opinion k at time t by

$$g_k(t) = \frac{h_k(t)}{\sum_{k'=1}^{K} h_{k'}(t)}.$$

Since our opinion dynamics model defines a stochastic process, we consider the problem of predicting the *expected share* of each opinion k at a given target time T, denoted by $\overline{g}_k(T)$. For solving this problem, we develop a method that effectively estimates the values of *value parameters* $w_1, \cdots, w_K$ from the observed data $\mathcal{D}_{T_0}$.

Simple Case Analysis

We analyze the effects of value parameters at the time t where all nodes have become active for an extreme case in which the network is complete, i.e., neighbors of each node cover the whole network. According to the previous work (e.g., (Sood and Redner 2005)), the expected share change $dg_k(t)$ can be calculated as follows:

$$\begin{aligned} dg_k(t) &= \frac{1}{|V|}(1 - g_k(t))\frac{g_k(t)w_k}{\sum_{k'=1}^{K} g_{k'}(t)w_{k'}} \\ &\quad -\frac{1}{|V|}g_k(t)\left(1 - \frac{g_k(t)w_k}{\sum_{k'=1}^{K} g_{k'}(t)w_{k'}}\right) \\ &= \frac{1}{|V|}\left(\frac{g_k(t)w_k}{\sum_{k'=1}^{K} g_{k'}(t)w_{k'}} - g_k(t)\right). \quad (1) \end{aligned}$$

Now, let k^* be the opinion with the highest value parameter such that $w_{k^*} > w_k$ for all the other opinion k $(k \neq k^*)$. Then, we can obtain the following inequality from Eq. (1) when $g_k(t) > 0$ for all k:

$$\begin{aligned} dg_{k^*}(t) &= \frac{g_{k^*}(t)w_{k^*}}{|V|\sum_{k=1}^{K} g_k(t)w_k}\left(1 - \sum_{k=1}^{K} g_k(t)\frac{w_k}{w_{k^*}}\right) \\ &> \frac{g_{k^*}(t)w_{k^*}}{|V|\sum_{k=1}^{K} g_k(t)w_k}\left(1 - \sum_{k=1}^{K} g_k(t)\right) = 0. \end{aligned}$$

Here note that $w_k/w_{k^*} < 1$ for $k \neq k^*$. Therefore, unless $g_{k^*}(t) = 0$, the opinion k^* is expected to finally prevail the others, regardless of its current share since the function $g_{k^*}(t)$ is expected to increase as time passes until each of the other opinion shares becomes 0. This result suggests that it is crucially important to accurately estimate the value parameter of each opinion from the observed data $\mathcal{D}_{T_0}$. Moreover, we can see that if the value parameters are uniform, any opinion can become a winner. These observations imply that the expected share prediction problem can be well-defined only when the opinion values are non-uniform. We conjecture that results will be similar for more realistic networks, although the above analysis is valid for a complete network.

Consensus Time Analysis

We further analyze the consensus time by using the above simple case. For simplicity, we assume that $w_k = w$ if $k \neq k^*$, i.e., the values of the other value parameters are the same. Let r be the ratio of the value parameters defined by $r = w/w_{k^*}$; then, by regarding $1/|V|$ as a time step dt (e.g., (Sood and Redner 2005)), we can obtain the following differential equation for $g_{k^*}(t)$ from Eq. (1):

$$\begin{aligned} \frac{dg_{k^*}(t)}{dt} &= \frac{g_{k^*}(t)}{r(1 - g_{k^*}(t)) + g_{k^*}(t)} - g_{k^*}(t) \\ &= \frac{(1 - r)g_{k^*}(t)(1 - g_{k^*}(t))}{r + (1 - r)g_{k^*}(t)}. \end{aligned}$$

From this differential equation, we can easily derive the following solution:

$$\frac{r}{1 - r}\log(g_{k^*}(t)) - \frac{1}{1 - r}\log(1 - g_{k^*}(t)) = t + C,$$

where C stands for a constant of integration. Figure 2 shows examples of expected share curves based on the above solution with different ratios of the value parameters, where the ratio r is set to $r = 1 - 2^{-a}$ $(a = 1, 2, 3, 4, 5)$, and each curve is plotted from $t = 0$ by assuming $g_{k^*}(0) = 0.01$ until $t = T$ that satisfies $g_{k^*}(T) = 0.99$. From this figure, we can see that the consensus time is quite short when the ratio r is small, while it takes somewhat longer when the ratio r approaches to 1. More importantly, this result indicates that the consensus time of our model is extremely short even

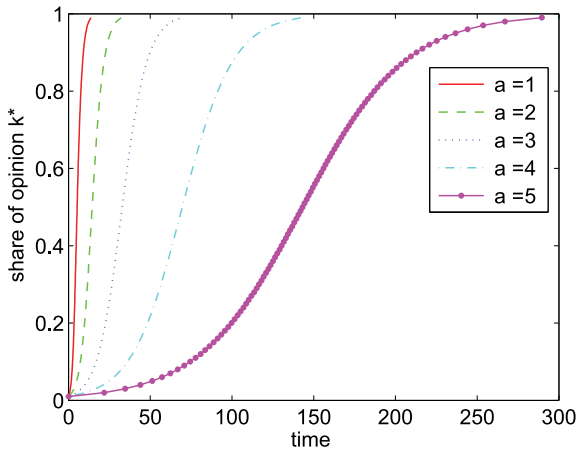


Figure 2: Examples of expected share curves.

when the ratio r is close to 1, compared with the basic voter model studied in previous work (e.g., (Even-Dar and Shapria 2007)). Therefore, we consider that the voter model can be more practical by introducing the value parameters.

Learning Method

For a given observed opinion diffusion data $\mathcal{D}_{T_0}$, we focus on the competitive opinion diffusion data $\mathcal{C}_{T_0}$ defined by

$$\mathcal{C}_{T_0} = \{(v, t, f_t(v)) \in \mathcal{D}_{T_0}; |\{u \in \Gamma(v);\ f_t(u) \neq 0\}| \geq 2\}.$$

Then, from the evolution process of our model described in the previous section, we can obtain the following likelihood function[2]:

$$\mathcal{L}(\boldsymbol{w}; \mathcal{C}_{T_0}) = \log \prod_{(v,t,k)\in\mathcal{C}_{T_0}} \frac{n_k(t,v) w_k}{\sum_{k'=1}^{K} n_{k'}(t,v) w_{k'}}, \qquad (2)$$

where $\boldsymbol{w}$ stands for the K-dimensional vector of value parameters, i.e., $\boldsymbol{w} = (w_1, \cdots, w_K)$. Thus our estimation problem[3] is formulated as a maximization problem of the objective function $\mathcal{L}(\boldsymbol{w}; \mathcal{C}_{T_0})$ with respect to $\boldsymbol{w}$.

Note that the objective function $\mathcal{L}(\boldsymbol{w}; \mathcal{C}_{T_0})$ is invariant to positive scaling of the value parameter vector $\boldsymbol{w}$, and each value parameter w_k must be positive, as noted earlier. In order to formulate our maximization problem as an unconstrained optimization problem, we reparameterize each value parameter w_k by using a $(K-1)$-dimensional vector $\boldsymbol{z} = (z_1, \cdots, z_{K-1})$ as follows:

$$w_k = \begin{cases} \exp(z_k) & \text{if } k < K, \\ 1 & \text{if } k = K. \end{cases} \qquad (3)$$

Namely, our estimation problem is formulated as an optimization problem of the objective function $\mathcal{L}_1(\boldsymbol{z}; \mathcal{C}_{T_0})$ $(= \mathcal{L}(\boldsymbol{w}; \mathcal{C}_{T_0}))$ with respect to $\boldsymbol{z}$.

[2]Introduction of $\mathcal{C}_{T_0}$ is simply to avoid $\log(0/0)$ and $\log 1$.

[3]The delay time parameter λ can also be a parameter, but it can simply be estimated by averaging the time intervals for each node, and thus excluded from the estimation problem. Estimating this parameter is not critical to the current problem because its value simply contributes to scaling the time unit.

In order to derive our learning algorithm, we consider the following probability that the node v adopts the opinion k $(k < K)$ at time t.

$$q_k(t,v) = \frac{n_k(t,v)\exp(z_k)}{n_K(t,v) + \sum_{k'=1}^{K-1} n_{k'}(t,v)\exp(z_{k'})} \qquad (4)$$

Then, we can obtain the first-order derivative (gradient vector element) of $\mathcal{L}_1(\boldsymbol{z}; \mathcal{C}_{T_0})$ with respect to z_i as follows:

$$\frac{\partial \mathcal{L}_1(\boldsymbol{z}; \mathcal{C}_{T_0})}{\partial z_i} = \sum_{(v,t,k)\in\mathcal{C}_{T_0}} (\delta_{k,i} - q_i(t,v)),$$

where $\delta_{k,i}$ is the Kronecker's delta. Similarly, we can obtain the second-order derivative (Hessian matrix element) with respect to z_i and z_j as follows:

$$\frac{\partial^2 \mathcal{L}_1(\boldsymbol{z}; \mathcal{C}_{T_0})}{\partial z_i \partial z_j} = \sum_{(t,v,k)\in\mathcal{C}_{T_0}} (q_i(t,v)q_j(t,v) - \delta_{i,j} q_i(t,v)).$$

Here note that the following quadratic form of the Hessian matrix is non-positive for an arbitrary $(K-1)$-dimensional non-zero vector $\boldsymbol{x} = (x_1, \cdots, x_{K-1})$,

$$\begin{aligned}
&\sum_{i,j=1}^{K-1} \frac{\partial^2 \mathcal{L}_1(\boldsymbol{z}; \mathcal{C}_{T_0})}{\partial z_i \partial z_j} x_i x_j \\
&= \sum_{(v,t,k)\in\mathcal{C}_{T_0}} \left(\left(\sum_{i=1}^{K-1} q_i(t,v) x_i \right)^2 - \sum_{i=1}^{K-1} q_i(t,v) x_i^2 \right) \\
&= -\sum_{(v,t,k)\in\mathcal{C}_{T_0}} \sum_{i=1}^{K-1} q_i(t,v) \left(x_i - \sum_{j=1}^{K-1} q_j(t,v) x_j \right)^2 \\
&\quad - \sum_{(v,t,k)\in\mathcal{C}_{T_0}} \left(1 - \sum_{i=1}^{K-1} q_i(t,v) \right) \left(\sum_{j=1}^{K-1} q_j(t,v) x_j \right)^2 \\
&\leq 0.
\end{aligned}$$

Thus we can guarantee that the solution of our problem is global optimal. Our implemetation employes a sandard Newton method. The algorithm of the proposed method is summarized below.

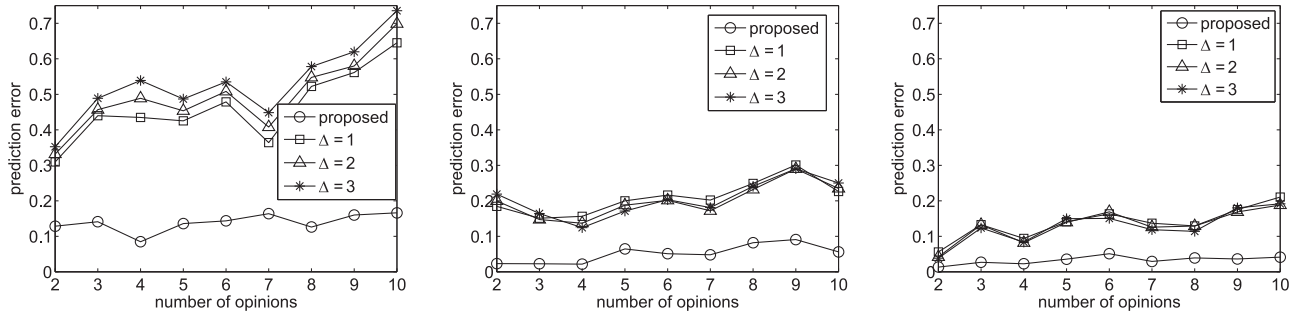
1. Initialize parameter vector $\boldsymbol{z}$ as $z_k = 0$ for $k = 1, \cdots, K-1$.

2. Calculate the gradient vector at the current parameter vector $\boldsymbol{z}$.

3. If the gradient vector is sufficiently small, i.e., $\sum_i (\partial \mathcal{L}_1(\boldsymbol{z}; \mathcal{C}_{T_0})/\partial z_i)^2 < \eta$, output the value parameters by using Eq. (3) then terminate. Otherwise, go to 4.

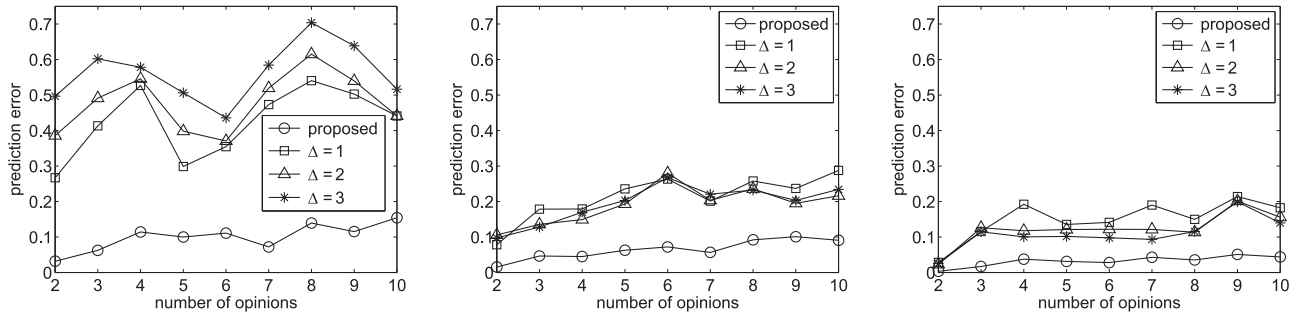
4. Calculate the Hessian matrix and its inverted matrix, and update the parameter vector $\boldsymbol{z}$ by multiplying the inverted matrix and the gradient vector, and return to 2.

Here η is a parameter for the termination condition. In our experiments, η is set to a sufficiently small number, i.e., $\eta = 10^{-12}$.



(a) $T_0 = 5, T = 30$ (b) $T_0 = 10, T = 30$ (c) $T_0 = 15, T = 30$

Figure 3: Results for share prediction in the blog network.



(a) $T_0 = 5, T = 30$ (b) $T_0 = 10, T = 30$ (c) $T_0 = 15, T = 30$

Figure 4: Results for share prediction in the Wikipedia network.

Experimental Evaluation

Network Datasets and Experimental Settings

We employed two datasets of large real networks used in (Kimura, Saito, and Motoda 2009), which are bidirectional connected networks and exhibit many of the key features of social networks. The first one is a trackback network of Japanese blogs and had $12,047$ nodes and $79,920$ directed links (the blog network). The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia, and had $9,481$ nodes and $245,044$ directed links (the Wikipedia network).

We varied $K = 2, 3, \cdots, 10$, and for each of them we predicted the expected share $\overline{g}_k(T)$ of opinion k ($k = 1, 2, \cdots, K$) for the observed data $\mathcal{D}_{T_0}$. We set $T = 30$, investigated the cases $T_0 = 5, 10, 15$, and selected the true value of each value parameter w_k from the interval $[0.5, 1.5]$ uniformly at random. We chose the top K nodes with respect to node degree ranking as the initial K nodes, and generated $\mathcal{D}_{T_0}$ by simulating the true model. After we have estimated the value of each w_k, we predicted the value of $\overline{g}_k(T)$ by simulating the model M times from $\mathcal{D}_{T_0}$ and taking their average, where we used $M = 100$. In fact, our preliminary experiments indicate that the result for $M = 100$ are not much different from those for $M = 1,000$ and $10,000$ in the blog and the Wikipedia networks. Note that the number of opinion updates amounts to tens of thousands for one instance of $\mathcal{D}_{T_0}$, and thus no overfitting problem arises.

Comparison Methods and Evaluation Measure

Given the observed data $\mathcal{D}_{T_0}$, we can simply apply a linear extrapolation for predicting the expected share of opinion k at a target time T, since we can naively speculate that the recent trend for each opinion continues. Thus, we consider predicting the values of $\overline{g}_1(T)$, $\cdots$, $\overline{g}_K(T)$, by estimating the value of the population $h_k(T)$ of opinion k at time T based on the linear extrapolation from the values of $h_k(T_0 - \Delta)$ and $h_k(T_0)$ for each k, where Δ is the parameter with $0 < \Delta \leq T_0$. We refer to this prediction method as the *naive linear method*. We evaluated the effectiveness of the proposed share prediction method by comparing it with the *naive linear method*.

Let $\widehat{g}_k(T)$ be the estimate of $\overline{g}_k(T)$ by a share prediction method. We measured the performance of the share prediction method by the prediction error $\mathcal{E}$ defined by

$$\mathcal{E} = \sum_{k=1}^{K} |\widehat{g}_k(T) - \overline{g}_k(T)|.$$

Experimental Results

Figures 3a, 3b, and 3c are the results for the blog network, and Figures 4a, 4b, and 4c for the Wikipedia network, where circles indicate the prediction errors of the proposed method,

and squares, triangles, and asterisks indicate the prediction errors of the *naive linear method* adopting $\Delta = 1$, $\Delta = 2$, and $\Delta = 3$, respectively. We conducted 10 trials varying the true values of value parameters for each K, and plotted the average of $\mathcal{E}$ over the 10 trials.

From these figures, we can see that the prediction error decreases as the observation time T_0 becomes longer and that the proposed method outperforms the *naive linear method* in every case. When $T_0 = 5$, the average prediction error of the proposed method was 0.139 for the blog network and 0.100 for the Wikipedia network, while that of the *naive method* was at least 0.465 and 0.424, respectively in case of $\Delta = 1$. When $T_0 = 15$, the average prediction error of the proposed method was 0.033 for the blog network and 0.032 for the Wikipedia network, while that of the *naive method* was at least 0.128 for the blog network and 0.110 for the Wikipedia network in case of $\Delta = 3$, which is comparable to those of the proposed method for $T_0 = 5$. Moreover, we observed that the proposed method accurately predicted the share at T even in the case that the share ranking at T_0 got reversed at the target time T as shown in Figure 1. This is attributed to the use of the estimated value parameters which take different values for different opinions, and is consistent with the aforementioned analysis on a complete network.

During the experiments we noticed that the time needed to reach the consensus gets longer when the difference between the largest and the second largest values of the value parameters is small. This can also be predicted by the consensus time analysis, i.e., considering the case where the highest two values are the same and the rest are also the same.

Consequently, we confirmed that the results of our theoretical analyses hold in real networks and that the proposed method outperforms the *naive linear method*. On average, the prediction error of the proposed method was about four times less for a given T_0. Besides, it achieved a comparable prediction accuracy in three times less observation time compared with the *naive linear method*.

Conclusion

We addressed the problem of how different opinions with different values spread over a social network and how their share changes over time in a machine learning setting using a variant of *voter model*, the value-weighted voter model with multiple opinions. The task is first to estimate the opinion values from the limited amount of observed data and the goal is to predict the expected opinion share at a future target time. We derived an algorithm that guarantees the global optimal solution for the opinion value estimation and showed using two real world social networks that the values are learnable from a small amount of observed data and the share prediction with use of the estimated values is satisfactorily accurate and outperforms the prediction by a simple linear extrapolation. Theoretical analysis for an extreme case where all the nodes are connected to each other (a complete network) revealed that the expected share prediction problem is well-defined only when the opinion values are non-uniform in which case the final consensus is winners-takes-all, i.e., the opinion with the highest value wins and all the others die, and when they are uniform, any opinion can be a winner. Our immediate future work is to validate the credibility of the *voter model* using available real opinion propagation data.

Acknowledgments This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-10-4053 and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

Agarwal, N., and Liu, H. 2008. Blogosphere: Research issues, tools, and applications. *SIGKDD Explorations* 10:18–31.

Castellano, C.; Munoz, M. A.; and Pastor-Satorras, R. 2009. Nonlinear q-voter model. *Physical Review E* 80:041129.

Chen, W.; Wang, Y.; and Yang, S. 2009. Efficient influence maximization in social networks. In *Proceedings of KDD 2009*, 199–208.

Crandall, D.; Cosley, D.; Huttenlocner, D.; Kleinberg, J.; and Suri, S. 2008. Feedback effects between similarity and sociai infiuence in online communities. In *Proceedings of KDD 2008*, 160–168.

Domingos, P., and Richardson, M. 2001. Mining the network value of customers. In *Proceedings of KDD 2001*, 57–66.

Even-Dar, E., and Shapria, A. 2007. A note on maximizing the spread of influence in social networks. In *Proceedings of WINE 2007*, 281–286.

Holme, P., and Newman, M. E. J. 2006. Nonequilibrium phase transition in the coevolution of networks and opinions. *Physical Review E* 74:056108.

Kempe, D.; Kleinberg, J.; and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *Proceedings of KDD 2003*, 137–146.

Kimura, M.; Saito, K.; Nakano, R.; and Motoda, H. 2010. Extracting influential nodes on a social network for information diffusion. *Data Mining and Knowledge Discovery* 20:70–97.

Kimura, M.; Saito, K.; and Motoda, H. 2009. Blocking links to minimize contamination spread in a social network. *ACM Transactions on Knowledge Discovery from Data* 3:9.

Leskovec, J.; Adamic, L. A.; and Huberman, B. A. 2007. The dynamics of viral marketing. *ACM Transactions on the Web* 1:5.

Liggett, T. M. 1999. *Stochastic interacting systems: contact, voter, and exclusion processes*. New York: Spriger.

Richardson, M., and Domingos, P. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of KDD 2002*, 61–70.

Sood, V., and Redner, S. 2005. Voter model on heterogeneous graphs. *Physical Review Letters* 94:178701.

Wu, F., and Huberman, B. A. 2008. How public opinion forms. In *Proceedings of WINE 2008*, 334–341.

Yang, H.; Wu, Z.; Zhou, C.; Zhou, T.; and Wang, B. 2009. Effects of social diversity on the emergence of global consensus in opinion dynamics. *Physical Review E* 80:046108.

Behavioral Analyses of Information Diffusion Models by Observed Data of Social Network

Kazumi Saito[1], Masahiro Kimura[2], Kouzou Ohara[3], and Hiroshi Motoda[4]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[3] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We investigate how well different information diffusion models explain observation data by learning their parameters and performing behavioral analyses. We use two models (CTIC, CTLT) that incorporate continuous time delay and are extension of well known Independent Cascade (IC) and Linear Threshold (LT) models. We first focus on parameter learning of CTLT model that is not known so far, and apply it to two kinds of tasks: ranking influential nodes and behavioral analysis of topic propagation, and compare the results with CTIC model together with conventional heuristics that do not consider diffusion phenomena. We show that it is important to use models and the ranking accuracy is highly sensitive to the model used but the propagation speed of topics that are derived from the learned parameter values is rather insensitive to the model used.

1 Introduction

The growth of Internet has enabled to form various kinds of large-scale social networks, through which a variety of information including innovation, hot topics and even malicious rumors can be propagated in the form of so-called "word-of-mouth" communications. Social networks are now recognized as an important medium for the spread of information, and a considerable number of studies have been made [1–5]. Widely used information diffusion models in these studies are the *independent cascade (IC)* [6–8] and the *linear threshold (LT)* [9, 10]. They have been used to solve such problems as the *influence maximization problem* [7, 11].

These two models focus on different information diffusion aspects. The IC model is sender-centered and an active node influences its inactive neighbors *independently* with diffusion probabilities assigned to links. On the other hand, the LT model is receiver-centered and a node is influenced by its active neighbors if the sum of their weights

exceeds the threshold for the node. Which model is more appropriate depends on the situation and selecting appropriate model is not easy. In order to study this problem, first of all, we need to know how different model behaves differently and how well or badly explain the observation data. Both models have parameters that need be specified in advance: diffusion probabilities for the IC model, and weights for the LT model. However, their true values are not known in practice. This poses yet another problem of estimating them from a set of information diffusion results that are observed as time-sequences of influenced (activated) nodes. To the best of our knowledge, there are only a few methods that can estimate the parameter values for the IC models and its variant that incorporates continuous time delay (referred to as the CTIC model) [3, 12, 13], but none for the LT model.

With this background, we first propose a novel method of learning the parameter values of a variant of the LT model that incorporates continuous time delay, similar to the CTIC model. We refer to this model as the CTLT model. It is indispensable to be able to cope with continuous time delay to do realistic analyses of information diffusion because, in the real world, information propagates along the continuous time axis, and time-delays can occur during the propagation. Thus, the proposed method has to estimate not only the weight parameters but also the time-delay parameters from the observed data. Incorporating time-delay makes the time-sequence observation data structural. In order to exploit this structure, we introduce an objective function that rigorously represents the likelihood of obtaining such observed data sequences under the CTLT model on a given network, and obtain parameter values that maximize this function by deriving parameter update EM algorithm. Next, we experimentally analyze how different models affect the information diffusion results differently by applying the proposed method to two tasks and comparing the results with the method which we already developed with the CTIC model [13]. The first task is ranking influential nodes in a social network, and we show that ranking is highly sensitive to the model used. We also show that the proposed method works well and can extract influential nodes more accurately than the well studied conventional four heuristic methods that do not take diffusion phenomena explicitly. The second task is the behavioral analysis of topic propagation on a real world blog data. We show that both model well capture the propagation phenomena on different topics at this level of abstract characterization.

2 Proposed Method

2.1 Information Diffusion Model

For a given directed network (or equivalently graph) $G = (V, E)$, let V be a set of nodes (or vertices) and E a set of links (or edges), where we denote each link by $e = (v, w) \in E$ and $v \neq w$, meaning there exists a directed link from a node v to a node w. For each node v in the network G, we denote $F(v)$ as a set of child nodes of v as follows: $F(v) = \{w; (v, w) \in E\}$. Similarly, we denote $B(v)$ as a set of parent nodes of v as follows: $B(v) = \{u; (u, v) \in E\}$. We define the LT model. In this model, for every node $v \in V$, we specify a *weight* ($\omega_{u,v} > 0$) from its parent node u in advance such that $\sum_{u \in B(v)} \omega_{u,v} \leq 1$. The diffusion process from a given initial active set S proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is

chosen uniformly at random from the interval $[0, 1]$. At time-step t, an inactive node v is influenced by each of its active parent nodes, u, according to weight $\omega_{u,v}$. If the total weight from active parent nodes of v is at least threshold θ_v, that is, $\sum_{u \in B_t(v)} \omega_{u,v} \geq \theta_v$, then v will become active at time-step $t+1$. Here, $B_t(v)$ stands for the set of all the parent nodes of v that are active at time-step t. The process terminates if no more activations are possible. Next, we extend the LT model so as to allow continuous-time delays, and refer to the extended model as the *continuous-time linear threshold (CTLT) model*. In the CTLT model, in addition to the weight set $\{\omega_{u,v}\}$, we specify real values r_v with $r_v > 0$ in advance for each node $v \in V$. We refer to r_v as the *time-delay parameter* on node v. Note that r_v depends only on v, which means that it is the node v's decision when to receive the information once the activation condition has been satisfied. The diffusion process unfolds in continuous-time t, and proceeds from a given initial active set S in the following way. Suppose that the total weight from active parent nodes of v became at least threshold θ_v at time t for the first time. Then, v will become active at time $t + \delta$, where we choose a delay-time δ from the exponential distribution with parameter r_v. Further, note that even though some other non-active parent nodes of v become active during the time period between t and $t + \delta$, the activation time of v, $t + \delta$, still remains the same. The other diffusion mechanisms are the same as the LT model.

For an initial active node v, let $\varphi(v)$ denote the number of active nodes at the end of the random process for the CTLT model. Note that $\varphi(v)$ is a random variable. Let $\sigma(v)$ denote the expected value of $\varphi(v)$. We call $\sigma(v)$ the *influence degree* of v for the CTLT model.

2.2 Learning problem

For the sake of technical convenience, we introduce a slack weight $\omega_{v,v}$ for each node $v \in V$ so as to be $\omega_{v,v} + \sum_{u \in B(v)} \omega_{u,v} = 1$. Here note that such a slack weight $\omega_{v,v}$ never contributes to the activation of v. We define the parameter vectors $\boldsymbol{r}$ and $\boldsymbol{\omega}$ by $\boldsymbol{r} = (r_v)_{v \in V}$ and $\boldsymbol{\omega} = (\omega_{u,v})_{(u,v) \in E}$. In practice, their true values are not available. Thus, we must estimate them from past information diffusion histories.

We consider an observed data set of M independent information diffusion results, $\mathcal{D}_M = \{D_m;\ m = 1, \cdots, M\}$. Here, each D_m is a time-sequence of active nodes in the mth information diffusion result (called mth result, hereafter for simplicity),

$$D_m = \langle D_m(t);\ t \in \mathcal{T}_m \rangle, \qquad \mathcal{T}_m = \langle t_m, \cdots, T_m \rangle,$$

where $D_m(t)$ is the set of all the nodes that have first become active at time t, and $\mathcal{T}_m$ is the observation-time list; t_m is the initial observed time and T_m is the final observed time. We assume that for any active node v in the mth result, there exits some $t \in \mathcal{T}_m$ such that $v \in D_m(t)$. Let $t_{m,v}$ denote the time at which node v has become active in the mth result, i.e., $v \in D_m(t_{m,v})$. For any $t \in \mathcal{T}_m$, we set

$$C_m(t) = \bigcup_{\tau \in \mathcal{T}_m \cap \{\tau;\, \tau < t\}} D_m(\tau)$$

Note that $C_m(t)$ is the set of nodes that had become active before time t in the mth result. We also interpret D_m as referring to the set of all the active nodes in the mth result for convenience sake. The problem is to estimate the values of $\boldsymbol{r}$ and $\boldsymbol{\omega}$ from $\mathcal{D}_M$.

2.3 Likelihood function

For the learning problem described above, we derive the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\omega}$ in a rigorous way to use as our objective function. Here note that for each node v, since a threshold θ_v is chosen uniformly at random from the interval $[0, 1]$, we can regard each weight $\omega_{*,v}$ as a multinomial probability, namely, $\omega_{v,v} + \sum_{u \in B(v)} \omega_{u,v} = 1$.

Suppose that a node v became active at time $t_{m,v}$ for the mth result. Then, we know that the total weight from active parent nodes of v became at least threshold θ_v at the time when one of these active parent nodes, $u \in B(v) \cap C_m(t_{m,v})$, became first active. However, in case of $|B(v) \cap C_m(t_{m,v})| > 1$, there is no way of exactly knowing the actual node due to the continuous time-delay. Suppose that a node v was actually activated when a node $\zeta \in B(v) \cap C_m(t_{m,v})$ became activated. Then θ_v is between $\sum_{u \in B(v) \cap C_m(t_{m,\zeta})} \omega_{u,v}$ and $\omega_{\zeta,v} + \sum_{u \in B(v) \cap C_m(t_{m,\zeta})} \omega_{u,v}$. Namely, the probability that θ_v is chosen from this range is $\omega_{\zeta,v}$. Here note that such events with respect to different active parent nodes are mutually disjoint. Thus, the probability density that the node v is activated at time $t_{m,v}$, denoted by $h_{m,v}$, can be expressed as

$$h_{m,v} = \sum_{u \in B(v) \cap C_m(t_{m,v})} \omega_{u,v} r_v \exp(-r_v(t_{m,v} - t_{m,u})). \tag{1}$$

Next, we consider any node $w \in V$ belonging to $\partial D_m = \{w; (v, w) \in E \wedge v \in C_m(T_m) \wedge w \notin D_m\}$ for the mth result. Let $g_{m,w}$ denote the probability that the node w is not activated by the node v within the observed time period $[t_m, T_m]$. Here we can naturally assume that each information diffusion process finished sufficiently earlier than the observed final time, i.e., $T_m \gg \max\{t; D_m(t) \neq \emptyset\}$. Thus, as $T_m \to \infty$, we obtain

$$g_{m,w} = 1 - \sum_{v \in B(w) \cap C_m(T_m)} \omega_{v,w}. \tag{2}$$

Therefore, by using equations (1), (2), and the independence properties, we can define the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\omega}$ by

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M) = \prod_{m=1}^{M} \left(\prod_{t \in \mathcal{T}_m} \prod_{v \in D_m(t)} h_{m,v} \prod_{w \in \partial D_m} g_{m,w} \right). \tag{3}$$

Thus, our problem is to obtain the time-delay parameter vector $\boldsymbol{r}$ and the diffusion parameter vector $\boldsymbol{\omega}$, which maximizes Equation (3). For this estimation problem, we can derive an estimation method based on the Expectation-Maximization algorithm in order to stably obtain its solutions, although we skip its derivation due to a space limitation.

2.4 Behavioral analysis

Thus far, we assumed that the time-delay and diffusion parameters can vary with respect to nodes and links but independent of the topic of information diffused. However, they may be sensitive to the topic.

Our method can cope with this by assigning a different m to a different topic, and placing a constraint that the parameters depends only on topics but not on nodes and links throughout the network G, that is $r_{m,v} = r_m$ and $\omega_{m,u,v} = q_m|B(v)|^{-1}$ for any node $v \in V$ or link $(u, v) \in E$. Here note that $0 < q_m < 1$ and $\omega_{v,v} = 1 - q_m$. This constraint is required because, without this, we have only one piece of observation for each (m, u, v) and there is no way to learn the parameters. Noting that we can naturally assume that people behave quite similarly for the same topic, this constraint should be acceptable. Under this setting, we can easily obtain the parameter update formulas. Using each pair of the estimated parameters, (r_m, q_m), we can analyze the behavior of people with respect to the topics of information, by simply plotting (r_m, q_m) as a point of 2-dimensional space (See Fig. 2 in Section 3.2).

3 Experiments

We applied the proposed learning method to two tasks to analyze how different models affect the information diffusion results differently and compared the results with the method which we already developed with the CTIC model [13]. First, we applied it to the problem of extracting influential nodes, and evaluated the performance of the CTLT model, i.e. parameter learning and influential node prediction, using the topologies of four large real network data. Next, we applied our method to behavioral analysis using a real world blog data based on the method described in section 2.4 and investigated how each topic spreads throughout the network.

3.1 Ranking Influential Nodes

Experimental Settings We employed four datasets of large real networks, which are all bidirectional connected networks. The first one is a trackback network of Japanese blogs used in [14] and had $12,047$ nodes and $79,920$ directed links (the blog network). The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia, also used in [14], and had $9,481$ nodes and $245,044$ directed links (the Wikipedia network). The third one is a network derived from the Enron Email Dataset [15] by extracting the senders and the recipients and linking those that had bidirectional communications and there were $4,254$ nodes and $44,314$ directed links (the Enron network). The fourth one is a co-authorship network used in [16] and had $12,357$ nodes and $38,896$ directed links (the coauthorship network).

Here, we assumed the simplest case where $\omega_{u,v} = q|B(v)|^{-1}$ and $r_v = r$ for any $u, v \in V$. One reason behind this assumption is that there is no need that the observation sequence data have to pass through every link at least once. This drastically reduces the amount of data necessary to learn the parameters. Then, our task is to estimate the values of q and r. The true value of q was decided to be set to 0.9 in order to achieve reasonably high influence degrees of nodes, and the true value of r was decided to be chosen from two values, one with a relatively high value $r = 2$ (a short time-delay case) and the other with a relatively low value $r = 1/2$ (a long time-delay case). The training data $\mathcal{D}_M$ in the learning stage was constructed by generating each D_m from a randomly selected initial active node $D_m(0)$ using the true CTLT model. We chose

Table 1: Parameter estimation accuracy by the proposed method.

Blog network			Wikipedia network			Enron network			Coauthorship network		
r^*	$\mathcal{E}_q$	$\mathcal{E}_r$	r^*	$\mathcal{E}_q$	$\mathcal{E}_r$	r^*	$\mathcal{E}_q$	$\mathcal{E}_r$	r^*	$\mathcal{E}_q$	$\mathcal{E}_r$
2	0.024	0.060	2	0.015	0.028	2	0.013	0.031	2	0.023	0.043
1/2	0.017	0.012	1/2	0.016	0.007	1/2	0.011	0.004	1/2	0.024	0.011

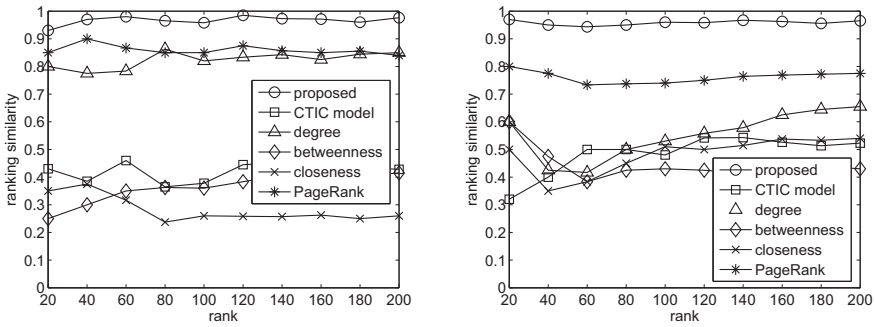
$T_m = \infty$ and used $M = 100$. We repeated the same experiment for each network five times independently.

We measure the influence of node v by the influence degree $\sigma(v)$ for the CTLT model that has generated $\mathcal{D}_M$. We compared the result of the high ranked influential nodes for the true CTLT model predicted by the proposed method with four heuristics widely used in social network analysis and the CTIC model based method [13]. The four heuristics are the same as those used in [13], "degree centrality", "closeness centrality", "betweenness centrality", and "authoritativeness" . The first three heuristics are commonly used as influence measure in sociology [17]. The authoritativeness is obtained by the "PageRank" method [18] which is a well known method for identifying authoritative or influential pages in a hyperlink network of web pages [5]. The CTIC model based method employs the CTIC model as the information diffusion model[13], where we learn the parameters of the CTIC model from the observed data $\mathcal{D}_M$, and rank nodes according to the influence degrees based on the learned model.

Experimental Results First, we examined the performance of estimating parameters by the proposed method. Let q^* and r^* denote the true values of q and r, respectively. Let $\hat{q}$ and $\hat{r}$ be the values of q and r estimated by the proposed method, respectively. We evaluated the parameter estimation accuracy by the errors $\mathcal{E}_q = |q^* - \hat{q}|$ and $\mathcal{E}_r = |r^* - \hat{r}|$. Table 1 shows the average values of $\mathcal{E}_q$ and $\mathcal{E}_r$ of five trials. We observe that the estimated values were close to the true values. The results demonstrate the effectiveness of the proposed method.

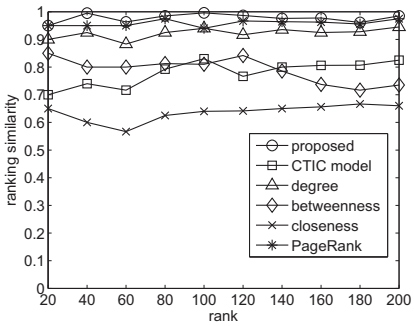
Next, in terms of extracting influential nodes from the network $G = (V, E)$, we evaluated the performance of the ranking methods mentioned above by the *ranking similarity* $\mathcal{F}(k) = |L^*(k) \cap L(k)|/k$ within the rank $k(> 0)$, where $L^*(k)$ and $L(k)$ are the true set of top k nodes and the set of top k nodes for a given ranking method, respectively. We focused on the performance for high ranked nodes since we are interested in extracting influential nodes. Figure 1 shows the results in the case of $r^* = 2$ for the blog, the Wikipedia, the Enron, and the coauthorship networks, respectively. For the proposed and the CTIC model methods, we plotted the average value of $\mathcal{F}(k)$ at k for five experimental results stated earlier. The results in the case of $r^* = 1/2$ for the proposed and the CTIC model methods were very similar to those in the case of $r^* = 2$. We see that the proposed method gives better results than the other methods for these networks, demonstrating the effectiveness of our proposed learning method. We also observe that the CTIC model method does not work well for predicting the high ranked influential nodes for the CTLT model for the problem setting we employed.

[5] As for the jump parameter ε of PageRank, we used a typical setting of $\varepsilon = 0.15$.

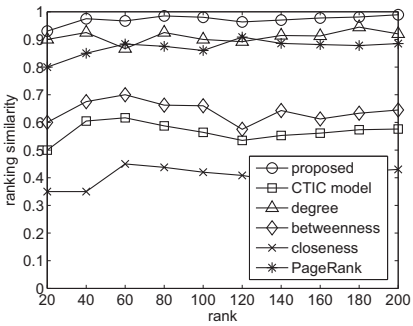


(a) Blog network

(b) Wikipedia network



(c) Enron network



(d) Coauthorship network

Fig. 1: Performance comparison in extracting influential nodes in the case of $r^* = 2$.

3.2 Behavioral Analysis of Real World Blog Data

Experimental Settings To compare the result by the proposed method with that by the CTIC model based method [13], we used the same real blogroll network as [13], which was generated from the database of a blog-hosting service in Japan called *Doblog* [6]. In the network, bloggers are connected to each other and we assume that topics propagate from blogger x to another blogger y when there is a blogroll link from y to x because this means that y is a reader of the blog of x. In addition, according to [19], it is supposed that a topic is represented as a URL which can be tracked down from blog to blog. We used the same propagation sequences of 172 URLs as [13] for this analysis, each of which is longer than 10 time steps. Please refer to [13] for more detailed description of the network generation and URL sequences.

Experimental Results We ran the experiments for each identified URL and obtained the corresponding parameters q and r. Figure 2 is a plot of the results for the major

[6] Doblog(`http://www.doblog.com/`), provided by NTT Data Corp. and Hotto Link, Inc.

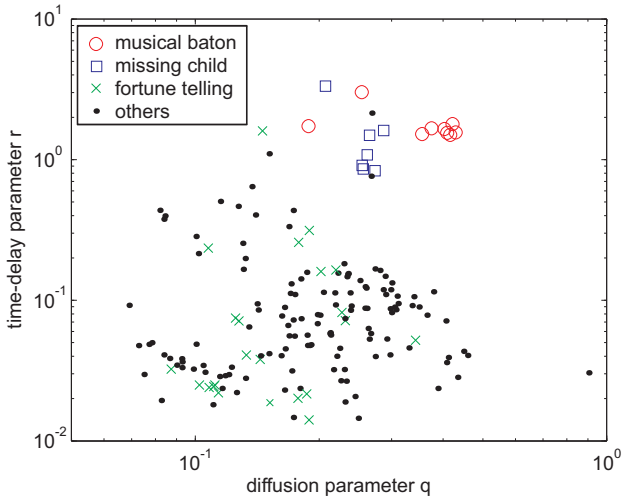


Fig. 2: Results for the Doblog database.

URLs. The horizontal axis is the diffusion parameter q and the vertical axis is the delay parameter r. The latter is normalized such that $r = 1$ corresponds to a delay of one day, meaning $r = 0.1$ corresponds delay of 10 days. In general, from this result, it can be said that the proposed method can extract characteristic properties of certain topics reasonably well only from the observation data. We only explain three URLs that exhibit some interesting propagation properties. The circle is a URL that corresponds to the musical baton which is a kind of telephone game on the Internet. It is shown that this kind of message propagates quickly (less than one day on the average) with a good chance (one out of 25 to 100 persons responds). This is probably because people are easily interested in and influenced by this kind of message passing. The square is a URL that corresponds to articles about a missing child. This also propagates quickly with a meaningful probability (one out of 80 persons responds). This is understandable considering the urgency of the message. The cross is a URL that corresponds to articles about fortune telling. Peoples responses are diverse. Some responds quickly (less than one day) and some late (more than one month after), and they are more or less uniformly distributed. The diffusion probability is also nearly uniformly distributed. This reflects that each individual's interest is different on this topic. The dot is a URL that corresponds to one of the other topics (not necessarily the same).

4 Discussion

With the addition of the proposed method, we now have ways to compare the diffusion process with respect to two models (the CTIC model and the CTLT model) for the same observed dataset. Being able to learn the parameters of these models enable us to analyze the diffusion process more precisely. Comparing the results bring us deeper

insights into the relation between models and information diffusion processes. Hence, we consider the contribution of the proposed method is significant.

Indeed, we obtained two interesting insights through the comparative experiments in the previous section. The first one comes from the results of ranking influential nodes, in which the ranking accuracy by the proposed method was better than those by the conventional heuristics, which was sort of expected, but the accuracy by the CTIC method was not, which is rather surprising. This means that the ranking results that involve detailed probabilistic simulation is very sensitive to the underlying model assumed to generate the observed data. When the underlying model is the CTIC and the data is generated by this model, the model learned outperforms these heuristics [13]. In other words, it is very important to select an appropriate model for the analysis of information diffusion from which the data has been generated. However, this is a very hard problem in reality. The second one comes from the results of the behavior analysis of topic propagation. The pattern shown in Fig.2 was very similar to that by the CTIC method shown in [13]. Regardless of the model used, in both results, the parameters for the topics that actually propagated quickly/slowly in observation converged to the values that enable them to propagate quickly/slowly on the model. Namely, we can say that the difference of models used has little influence on the relative difference of topic propagation property which indeed strongly depends on topic itself. Both models are well defined and can explain this property at this level of abstraction. However, we have to carefully choose a model at least when solving such problems as the influence maximization problem [7, 11], a problem at a more detailed level.

5 Conclusion

We considered the problem of analyzing information diffusion process in a social network using two kinds of information diffusion models, incorporating continuous time delay, the CTIC model and the CTLT model, and investigated how the results differ according to the model used. To this end, we proposed a novel method of learning the parameters of the CTLT model from the observed data, and experimentally confirmed that it works well on real world datasets. We also obtained the following two important observations through the experiments for the two tasks. One is that in learning the information diffusion parameters of nodes and links, the learning results are highly sensitive to the model used. The other is that in analyzing the topic-oriented characteristics such as the propagation speed of each topic, using different models has little influence on the analysis results. These two contrasting observations may hold only for well-defined diffusion models such as the CTIC and CTLT models. These findings would help us consider whether we should select a model carefully, or not. In practice, as there are numerous factors that affects the information diffusion process, it is difficult to select an appropriate model in a more realistic setting. This model selection is our future work.

Acknowledgment

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory

under Grant No. AOARD-08-4027, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Physical Review E **66** (2002) 035101
2. Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45** (2003) 167–256
3. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. SIGKDD Explorations **6** (2004) 43–52
4. Domingos, P.: Mining social networks for viral marketing. IEEE Intelligent Systems **20** (2005) 80–82
5. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06). (2006) 228–237
6. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters **12** (2001) 211–223
7. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003). (2003) 137–146
8. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data **3** (2009) 9:1–9:23
9. Watts, D.J.: A simple model of global cascades on random networks. Proceedings of National Academy of Science, USA **99** (2002) 5766–5771
10. Watts, D.J., Dodds, P.S.: Influence, networks, and public opinion formation. Journal of Consumer Research **34** (2007) 441–458
11. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07). (2007) 1371–1376
12. Saito, K., Kimura, M., Nakano, R., Motoda, H.: Finding influential nodes in a social network from information diffusion data. In: Proceedings of the International Workshop on Social Computing and Behavioral Modeling (SBP09). (2009) 138–145
13. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Learning continuous-time information diffusion model for social behavioral data analysis. In: Proceedings of the 1st Asian Conference on Machine Learning (ACML2009). (2009) 322–337
14. Kimura, M., Saito, K., Motoda, H.: Minimizing the spread of contamination by blocking links in a network. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08). (2008) 1175–1180
15. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: Proceedings of the 2004 European Conference on Machine Learning (ECML'04). (2004) 217–226
16. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435** (2005) 814–818
17. Wasserman, S., Faust, K.: Social network analysis. Cambridge University Press, Cambridge, UK (1994)
18. Brin, S., L.Page: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems **30** (1998) 107–117
19. Adar, E., Adamic, L.A.: Tracking information epidemics in blogspace. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05). (2005) 207–214

Extracting Influential Nodes on a Social Network for Information Diffusion

Masahiro Kimura, Kazumi Saito, Ryohei Nakano, and Hiroshi Motoda

Abstract

We address the combinatorial optimization problem of finding the most influential nodes on a large-scale social network for two widely-used fundamental stochastic diffusion models. The past study showed that a greedy strategy can give a good approximate solution to the problem. However, a conventional greedy method faces a computational problem. We propose a method of efficiently finding a good approximate solution to the problem under the greedy algorithm on the basis of bond percolation and graph theory, and compare the proposed method with the conventional method in terms of computational complexity in order to theoretically evaluate its effectiveness. The results show that the proposed method is expected to achieve a great reduction in computational cost. We further experimentally demonstrate that the proposed method is much more efficient than the conventional method using large-scale real-world networks including blog networks.

Keywords

Social network analysis, Information diffusion model, Influence maximization problem, Bond percolation

Authors' Addresses:

Masahiro Kimura
Department of Electronics and Informatics
Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp

Kazumi Saito
School of Administration and Informatics
University of Shizuoka
Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp

Ryohei Nakano
Department of Computer Science
Chubu University
Aichi 487-8501, Japan
nakano@cs.chubu.ac.jp

Hiroshi Motoda
Institute of Scientific and Industrial Research
Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

1 Introduction

The rise of the Internet and the World Wide Web has enabled us to investigate large-scale social networks, and there has been growing interest in social network analysis (Newman, 2001; McCallum et al., 2005; Leskovec et al., 2006). Here, a social network is the network of relationships and interactions among social entities such as individuals, groups of individuals, and organizations. Examples include blog networks, collaboration networks, and email networks.

The social network of interactions within a group of individuals plays a fundamental role in the spread of information, ideas, and innovations. In fact, a piece of information, such as the URL of a website that provides a new valuable service, can spread from one individual to another through the social network in the form of "word-of-mouth" communication. For example, the information of free email services such as Microsoft's Hotmail and Google's Gmail could spread largely through email networks. Thus, when we plan to market a new product, promote an innovation, or spread a new topic among a group of individuals, we can exploit social network effects. Namely, we can *target* a small number of influential individuals (e.g., giving free samples of the product, demonstrating the innovation, or offering the topic), and trigger a cascade of influence by which friends will recommend the product, promote the innovation, or propagate the topic to other friends. In this way, we can spread decisions in adopting the product, the innovation, or the topic through the social network from a small set of initial adopters to many individuals. Therefore, given a social network represented by a directed graph, a positive integer k, and a probabilistic model for the process by which a certain information spreads through the network, it is an important research issue in terms of sociology and *viral marketing* to find such a target set A_k^* of k nodes that maximizes the expected number of adopters of the information if A_k^* initially adopts it (Domingos and Richardson, 2001; Richardson and Domingos, 2002; Kempe et al, 2003; Kempe et al., 2005). Here, the expected number of nodes influenced by a target set is referred to as its *influence degree*, and this combinatorial optimization problem is called the *influence maximization problem* of size k.

Kempe et al. (2003) studied the influence maximization problem for two widely-used fundamental information diffusion models, the *independent cascade (IC) model* (Goldenberg, 2001; Kempe et al., 2003; Gruhl et al., 2004) and the *linear threshold (LT) model* (Watts, 2002; Kempe et al., 2003). They experimentally showed on large collaboration networks that for the influence maximization problem under the IC and LT models, the greedy algorithm significantly outperforms the high-degree and centrality heuristics that are commonly used in the sociology literature. Here, the high-degree heuristic chooses nodes in order of decreasing degrees, and the centrality heuristic chooses nodes in order of increasing average distance to other nodes in the

network. Moreover, they mathematically proved a performance guarantee of the greedy algorithm under these information diffusion models (i.e., the IC and LT models) by using an analysis framework based on submodular functions.

For the influence maximization problem of size k, the greedy algorithm iteratively finds a target set A_k of k nodes from the target set A_{k-1} of $k-1$ nodes that it has already found. Thus, it requires a method of computing all the *marginal influence degrees* of a given set A of nodes in the network. Here, for any node v that does not belong to A, the influence degree of target set $A \cup \{v\}$ is referred to as the *marginal influence degree* of A at v. However, it is an open question to compute influence degrees exactly by an efficient method, and therefore, the conventional method had to obtain good estimates for influence degrees by simulating the random process of the information diffusion model (i.e., the IC or LT model) many times (Kempe et al., 2003). Solving the influence maximization problem under the greedy algorithm needed a large amount of computation for large-scale networks.

In this paper, for the IC and LT models, we propose a method of efficiently estimating all the marginal influence degrees of a given set of nodes on the basis of bond percolation and graph theory, and apply it to approximately solving the influence maximization problem under the greedy algorithm. In order to theoretically evaluate the effectiveness of the proposed method for solving the influence maximization problem, we compare the proposed method with the conventional method in terms of computational complexity, and show that the proposed method is expected to achieve a large reduction in computational cost. Further, using large-scale real networks including blog networks, we experimentally demonstrate that the proposed method is much more efficient than the conventional method. Finally, we discuss some related work, and describe the conclusion.

2 Definitions

We examine the influence maximization problem on a network represented by a directed graph $G = (V, E)$ for the IC and LT models. Here, V and E are the sets of all the nodes and links in the network, respectively. Let N and L be the numbers of elements of V and E, respectively.

We first recall some basic notions from graph theory. Next, we define the IC and LT models on G according to the work of Kempe et al. (2003). Last, we give a mathematical definition of the influence maximization problem.

2.1 Graphs

We consider a directed graph $G = (V, E)$. If there is a directed link (u, v) from node u to node v, node v is called a *child node* of node u and node u is called a *parent node* of node v. For any $v \in V$, let $\Gamma(v)$ denote the set of all

the parent nodes of v. For a subset V' of V, graph $G' = (V', E')$ is called the *induced graph* of G to V' if $E' = E \cap (V' \times V')$.

We call $(u_0, \cdots, u_\ell)$ a *path* from node u_0 to node u_ℓ if we have $(u_{i-1}, u_i) \in E$, $(i = 1, \cdots, \ell)$. We say that node u can *reach* node v or node v is *reachable* from node u if there is a path from node u to node v. For a node v of the graph G, we define $F(v; G)$ to be the set of all the nodes that are reachable from v, and define $B(v; G)$ to be the set of all the nodes that can reach v. For any $A \subset V$, we set

$$F(A; G) = \bigcup_{v \in A} F(v; G), \ B(A; G) = \bigcup_{v \in A} B(v; G).$$

A *strongly connected component (SCC)* of G is a maximal subset C of V such that for all u, $v \in C$ there is a path from u to v. For a node v of G, we define $SCC(v; G)$ to be the SCC that contains v.

2.2 Information Diffusion Models

We consider mathematically modeling the spread of certain information through a social network $G = (V, E)$. In the IC and LT models, the following assumptions are made:

- A node is called *active* if it has adopted the information.
- The state of a node is either *active* or *inactive*.
- Nodes can switch from being inactive to being active, but cannot switch from being active to being inactive.
- The spread of the information through the network G is represented as the spread of active nodes on G.
- Given an initial set A of active nodes, we suppose that the nodes in A first become active and all the other nodes remain inactive at time-step 0.
- The diffusion process of active nodes unfolds in discrete time-steps $t \geq 0$.

2.2.1 Independent Cascade Model

First, we define the *independent cascade (IC) model*. In this model, we specify a real value $p_{u,v} \in [0, 1]$ for each directed link (u, v) in advance. Here, $p_{u,v}$ is referred to as the *propagation probability* through link (u, v). When an initial set A of active nodes is given, the diffusion process of active nodes proceeds according to the following randomized rule. When node u first becomes active at time-step t, it is given a single chance to activate

each of its currently inactive child nodes v, and succeeds with probability $p_{u,v}$. If u succeeds, then v will become active at time-step $t+1$. Here, if v has multiple parent nodes that become active at time-step t for the first time, then their activation attempts are sequenced in an arbitrary order, but performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

For an initial active set A ($\subset V$), let $\varphi(A)$ denote the number of active nodes at the end of the random process for the IC model. Note that $\varphi(A)$ is a random variable. Let $\sigma(A)$ denote the expected value of $\varphi(A)$. We call $\sigma(A)$ the *influence degree* of A.

2.2.2 Linear Threshold Model

Next, we define the *linear threshold (LT) model*. In this model, for any node $v \in V$, we in advance specify a *weight* $w_{u,v}$ (> 0) from its parent node u such that

$$\sum_{u \in \Gamma(v)} w_{u,v} \leq 1.$$

When an initial set A of active nodes is given, the diffusion process of active nodes proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0,1]$. At time-step t, an inactive node v is influenced by each of its active parent nodes u according to weight $w_{u,v}$. If the total weight from active parent nodes of v is at least threshold θ_v, that is,

$$\sum_{u \in \Gamma_t(v)} w_{u,v} \geq \theta_v,$$

then v will become active at time-step $t+1$. Here, $\Gamma_t(v)$ stands for the set of parent nodes of v that are active at time-step t. The process terminates if no more activations are possible.

Note that the threshold θ_v models the tendency of node v to adopt the information when its parent nodes do. Note also that the LT model is a probabilistic model associated with the uniform distribution on $[0,1]^N$. Further note that in the LT model it is the node thresholds that are random, while in the IC model it is the propagations through links that are random. Suppose that A is an initial set of active nodes. We define a random variable $\varphi(A)$ by the number of active nodes at the end of the random process for the LT model. Let $\sigma(A)$ denote the expected value of $\varphi(A)$. We call $\sigma(A)$ the *influence degree* of A. Note that these notations are the same as those for the IC model.

2.3 Influence Maximization Problem

We mathematically define the influence maximization problem on a network $G = (V, E)$ under the IC and LT models. Let k be a positive integer with $k < N$.

The *influence maximization problem* on G of size k is defined as follows: Find a set A_k^* of k nodes to target for initial activation such that $\sigma(A_k^*) \geq \sigma(S)$ for any set S of k nodes, that is, find

$$A_k^* \ = \ \mathrm{argmax}_{A \in \{S \subset V;\, |S|=k\}} \, \sigma(A), \tag{1}$$

where $|S|$ stands for the number of elements of set S.

3 Conventional Method

Kempe et al. (2003) showed the effectiveness of the greedy algorithm for the influence maximization problem under the IC and LT models. In this section, we introduce the greedy algorithm, and describe the conventional method for solving the influence maximization problem under the greedy algorithm. We, then, consider evaluating the computational complexity for the conventional method.

3.1 Greedy Algorithm

We approximately solve the influence maximization problem by the following greedy algorithm:

(G1) Set $A \leftarrow \emptyset$.

(G2) **for** $i = 1$ to k **do**

(G3) Choose a node $v_i \in V$ maximizing $\sigma(A \cup \{v\})$, $(v \in V \setminus A)$.

(G4) Set $A \leftarrow A \cup \{v_i\}$.

(G5) **end for**

Let A_k denote the set of k nodes obtained by this algorithm. We refer to A_k as the *greedy solution* of size k. Then, it is known that

$$\sigma(A_k) \ \geq \ \left(1 - \frac{1}{e}\right) \sigma(A_k^*),$$

that is, the quality guarantee of A_k is assured (Kempe et al., 2003). Here, A_k^* is the exact solution defined by Equation (1).

To implement the greedy algorithm, we need a method for estimating all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A in Step (G3) of the algorithm.

3.2 Conventional Method for Estimating Marginal Influence Degrees

For Step (G3) of the greedy algorithm, the conventional method estimated all the marginal influence degrees $\{\sigma(A\cup\{v\});\, v \in V\backslash A\}$ of A in the following way (Kempe et al., 2003): First, a sufficiently large positive integer M is specified. For any $v \in V \setminus A$, the random process of the diffusion model (IC or LT model) is run from the initial active set $A \cup \{v\}$, and the number $\varphi(A \cup \{v\})$ of final active nodes is counted. Each $\sigma(A \cup \{v\})$ is estimated as the empirical mean obtained from M such simulations.

Namely, the conventional method independently estimated $\sigma(A \cup \{v\})$ for all $v \in V \setminus A$ as follows:

1. **for** $m = 1$ to M **do**
2. Compute $\varphi(A \cup \{v\})$.
3. Set $x_m \leftarrow \varphi(A \cup \{v\})$.
4. **end for**
5. Set $\sigma(A \cup \{v\}) \leftarrow (1/M) \sum_{m=1}^{M} x_m$.

Here, each $\varphi(A \cup \{v\})$ is computed as follows:

1. Set $H_0 \leftarrow A \cup \{v\}$.
2. Set $t \leftarrow 0$.
3. **while** $H_t \neq \emptyset$ **do**
4. Set $H_{t+1} \leftarrow \{$the activated nodes at time $t + 1\}$.
5. Set $t \leftarrow t + 1$.
6. **end while**
7. Set $\varphi(A \cup \{v\}) \leftarrow \sum_{j=0}^{t-1} |H_j|$

3.3 Computational Complexity of Conventional Method

We consider evaluating the computational complexity of solving the influence maximization problem. For this purpose, we introduce the notion of *examined nodes*. Here, an *examined node* is a node that is actually visited by tracing incoming or outgoing links on the graph in question for the method when all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A are estimated in Step (G3) of the greedy algorithm. In Section 4.4, we describe the reason why we investigate the examined nodes for evaluating the computational complexity.

The computational complexity of the conventional method is evaluated in terms of the expected number of examined nodes. In order to estimate $\sigma(A \cup \{v\})$, $(v \in V \setminus A)$, it is necessary for any $v \in V \setminus A$ to simulate M times the random process of the information diffusion model (IC or LT model) from the initial active set $A \cup \{v\}$ on graph G. For each simulation, the set of examined nodes are the same as the set of active nodes in the process. Thus, we can estimate that the expected number $\mathcal{C}_0$ of examined nodes for the conventional method is

$$\mathcal{C}_0 = M \sum_{v \in V \setminus A} \sigma(A \cup \{v\}). \tag{2}$$

4 Proposed Method

We propose a method for efficiently estimating all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A in Step (G3) of the greedy algorithm on the basis of bond percolation and graph theory, and evaluate the computational complexity, and compare it with that of the conventional method.

4.1 Bond Percolation

The IC and LT models are identified with *bond percolation models* which are defined below, and all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A are efficiently estimated by exploiting graph theoretic methods.

A *bond percolation* process on graph $G = (V, E)$ is the process in which each link of G is randomly designated either "occupied " or "unoccupied" according to some probability distribution. Here, in terms of information diffusion on a social network, occupied links represent the links through which the information propagates, and unoccupied links represent the links through which the information does not propagate. Let us consider the following set of L-dimensional vectors,

$$R_G = \left\{ r = (r_{u,v})_{(u,v) \in E} \in \{0,1\}^L \right\},$$

where L is the number of links in G. A bond percolation process on G is determined by a probability distribution $q(r)$ on R_G. Namely, for a random vector $r \in R_G$ drawn from $q(r)$, each link $(u, v) \in E$ is designated "occupied" if $r_{u,v} = 1$, and it is designated "unoccupied" if $r_{u,v} = 0$. Let E_r denote the set of all the occupied links for $r \in R_G$, and let G_r denote the graph (V, E_r). For each $r \in R_G$, we can consider the deterministic diffusion model $\mathcal{M}_r$ on G_r such that $F(A; G_r)$ becomes the final set of active nodes when A is an initial set of active nodes, where $F(A; G_r)$ is the set that is reachable from A on G_r (see, Section 2.1). By associating the diffusion model $\mathcal{M}_r$ on G_r with a probability distribution $q(r)$ on R_G, we define a stochastic diffusion model on G. We call this diffusion model the *bond percolation model* on G, and

refer to the probability distribution $q(r)$ on R_G as the *occupation probability distribution* of the bond percolation model.

We easily see that the IC model on G can be identified with the so-called *susceptible/infective/recovered (SIR) model* (Newman, 2003) for the spread of a disease on G, where the nodes that become active at time t in the IC model correspond to the infective nodes at time t in the SIR model. We recall that in the SIR model, an individual occupies one of the three states, "susceptible", "infected" and 'recovered", where a susceptible individual becomes infected with a certain probability when s/he is encountered an infected patient and subsequently recovers at a certain rate (see, Newman, 2003; Watts and Dodds, 2007). It is known that the SIR model on a network can be exactly mapped onto a bond percolation model on the same network (Grassberger, 1983; Newman, 2002; Kempe et al., 2003; Newman, 2003). Hence, we see that the IC model on G is equivalent to a bond percolation model on G, that is, these two models have the same probability distribution for the final set of active nodes given a target set. Here, for the IC model on G, the occupation probability distribution $q(r)$ of the corresponding bond percolation model is given by

$$q(r) = \prod_{(u,v)\in E} \left\{ (p_{u,v})^{r_{u,v}} (1-p_{u,v})^{1-r_{u,v}} \right\}, \quad (r \in R_G),$$

that is, each link (u, v) of G is independently declared to be "occupied" with probability $p_{u,v}$, where $p_{u,v}$ is the propagation probability through link (u, v) in the IC model.

On the other hand, Kempe et al. (2003) proved that the LT model on G can also be equivalent to a bond percolation model on G to derive the result that the influence degree function $\sigma(A)$ is submodular in the LT model. Here, for the LT model on G, the corresponding occupation probability distribution $q(r)$ is generated by declaring "occupied" and "unoccupied" links in the following way: For any $v \in V$, we pick at most one of the incoming links to v by selecting link (u, v) with probability $w_{u,v}$ and selecting no link with probability $1 - \sum_{u\in\Gamma(v)} w_{u,v}$. After this process, the picked links are declared to be "occupied" and the other links are declared to be "unoccupied". Here, $w_{u,v}$ is the weight of link (u, v) in the LT model. Specifically, $q(r)$ is described as follows:

$$q(r) \;=\; \prod_{v\in V} \prod_{u\in\Gamma(v)} \left\{ (w_{u,v})^{r_{u,v}} \left(1 - \sum_{u\in\Gamma(v)} w_{u,v} \right)^{\left(1-\sum_{u\in\Gamma(v)} r_{u,v}\right)} \right\},$$

where if $\sum_{u\in\Gamma(v)} w_{u,v} < 1$, $\sum_{u\in\Gamma(v)} r_{u,v} \le 1$ and if $\sum_{u\in\Gamma(v)} w_{u,v} = 1$, $\sum_{u\in\Gamma(v)} r_{u,v} = 1$.

4.2 Proposed Method for Estimating Marginal Influence Degrees

We present a method of estimating all the marginal influence degrees $\{\sigma(A \cup \{v\});\ v \in V \setminus A\}$ of A in Step (G3) of the greedy algorithm. As shown in the preceding section, the IC and LT models on G can be identified with the bond percolation models on G. Therefore, we have

$$\sigma(A \cup \{v\}) = \sum_{r \in R_G} q(r)\, |F(A \cup \{v\}; G_r)|$$

for any $v \in V \setminus A$, where $q(r)$ is the corresponding occupation probability distribution, and $F(A \cup \{v\}; G_r)$ stands for the set of all the nodes that are reachable from $A \cup \{v\}$ on graph G_r (see, Section 2.1).

We estimate $\{\sigma(A \cup \{v\});\ v \in V \setminus A\}$ in the following way: First, we specify a sufficiently large positive integer M. Next, we independently generate a set $\{r_1, \cdots, r_M\}$ of M sample vectors on R_G from the probability distribution $q(r)$; that is, independently generate a set $\{G_{r_m}; m = 1, \cdots, M\}$ of M graphs. For any $v \in V \setminus A$, we approximate $\sigma(A \cup \{v\})$ by

$$\sigma(A \cup \{v\}) \simeq \frac{1}{M} \sum_{m=1}^{M} |F(A \cup \{v\}; G_{r_m})|. \tag{3}$$

Thus, we estimate $\{\sigma(A \cup \{v\});\ v \in V \setminus A\}$ on the basis of Equation (3) as follows:

1. **for** $m = 1$ to M **do**
2. Generate graph G_{r_m}.
3. Compute $\{|F(A \cup \{v\}; G_{r_m})|;\ v \in V \setminus A\}$.
4. Set $x_{v,m} \leftarrow |F(A \cup \{v\}; G_{r_m})|$ for all $v \in V \setminus A$.
5. **end for**
6. Set $\sigma(A \cup \{v\}) \leftarrow (1/M) \sum_{m=1}^{M} x_{v,m}$ for all $v \in V \setminus A$.

In particular, we evaluate $\{|F(A \cup \{v\}; G_r)|;\ v \in V \setminus A\}$ for an arbitrary $r \in R_G$ by the following algorithm:

(E1) Find the subset $F(A; G_r)$ of V.

(E2) Set $|F(A \cup \{v\}; G_r)| \leftarrow |F(A; G_r)|$ for all $v \in F(A; G_r) \setminus A$.

(E3) Find the subset $V_r^A = V \setminus F(A; G_r)$ of V, and the induced graph G_r^A of G_r to V_r^A.

(E4) Set $U \leftarrow \emptyset$.

(E5) while $V_r^A \setminus U \neq \emptyset$ **do**

(E6) Pick a node $u \in V_r^A \setminus U$.

(E7) Find the subset $F(u; G_r^A)$ of V_r^A.

(E8) Find the subset $C(u; G_r^A) = B(u; G_r^A) \cap F(u; G_r^A)$ of $F(u; G_r^A)$.

(E9) Set $|F(A \cup \{v\}; G_r)| \leftarrow |F(u; G_r^A)| + |F(A; G_r)|$ for all $v \in C(u; G_r^A)$.

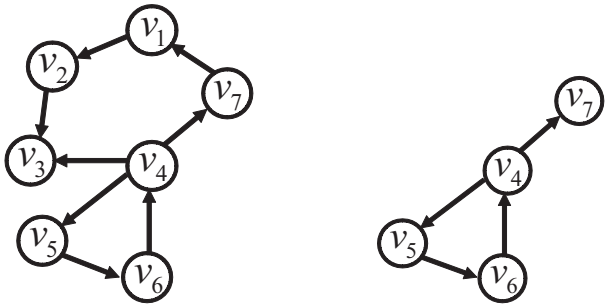
(E10) Set $U \leftarrow U \cup C(u; G_r^A)$.

(E11) end while

Now, we explain this algorithm. In Step (E1), we find the subset $F(A; G_r)$ that is reachable from A on graph G_r. In Step (E2), we use the fact that if $v \in F(A; G_r)$, the set $F(A \cup \{v\}; G_r)$ that is reachable from $A \cup \{v\}$ on G_r is equal to the set $F(A; G_r)$, and we simultaneously compute $|F(A \cup \{v\}; G_r)|$ for all $v \in F(A; G_r)$. In Step (E3), we find the subset $V_r^A = V \setminus F(A; G_r)$, and also find the induced graph G_r^A of graph G_r to V_r^A. In Steps (E4) to (E11), we use the fact that if $v \notin F(A; G_r)$, $|F(A \cup \{v\}; G_r)|$ is obtained by the sum of $|F(A; G_r)|$ and $|F(v; G_r^A)|$. This fact enables us to reduce the graph in question from G_r to G_r^A. We attempt to decompose graph G_r^A into its SCCs. In Step (E6), on graph G_r^A, we pick a node u that does not belong to the SCCs that we have already found. In Step (E7), we find the set $F(u; G_r^A)$ that is reachable from u on graph G_r^A. In Step (E8), we find the subset $C(u; G_r^A) = B(u; G_r^A) \cap F(u; G_r^A)$ of $F(u; G_r^A)$ by tracing backward all the links from u on the induced graph of G_r^A to $F(u; G_r^A)$. Note that the set $C(u; G_r^A)$ is equal to the SCC $SCC(u; G_r^A)$ that contains u. In Step (E9), we use the fact that $|F(v; G_r^A)| = |F(u; G_r^A)|$ if $v \in C(u; G_r^A)$, and simultaneously compute $|F(A \cup \{v\}; G_r)|$ for all $v \in C(u; G_r^A)$. We illustrate the flow of the algorithm in the following example:

Example: We consider the graph G_r shown in Figure 1a, where $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$. We set $A = \{v_1\}$. In this case, the process of the algorithm proceeds as follows.

In Step (E1), we find $F(A; G_r) = \{v_1, v_2, v_3\}$. In Step (E2), we find $|F(A \cup \{v_2\}; G_r)| = |F(A \cup \{v_3\}; G_r)| = 3$. In Step (E3), we find $V_r^A = \{v_4, v_5, v_6, v_7\}$ and G_r^A as shown in Figure 1b. In Step (E4), we set $U = \emptyset$. In Step (E5), we check $V_r^A \setminus U = \{v_4, v_5, v_6, v_7\} \neq \emptyset$. In Step (E6), we pick $v_4 \in V_r^A \setminus U$. In Step (E7), we find $F(v_4; G_r^A) = \{v_4, v_5, v_6, v_7\}$. In Step (E8), we find $C(v_4; G_r^A) = B(v_4; G_r^A) \cap F(v_4; G_r^A) = \{v_4, v_5, v_6\}$ in $F(v_4; G_r^A)$. In Step (E9), we find $|F(A \cup \{v_4\}; G_r)| = |F(A \cup \{v_5\}; G_r)| = |F(A \cup \{v_6\}; G_r)| = 7$. In Step (E10), we set $U = \{v_4, v_5, v_6\}$. In Step (E11), we return to Step (E5). In Step (E5), we check $V_r^A \setminus U = \{v_7\} \neq \emptyset$. In Step (E6), we pick $v_7 \in V_r^A \setminus U$. In Step (E7), we find $F(v_7; G_r^A) = \{v_7\}$. In Step (E8), we find $C(v_7; G_r^A) = \{v_7\}$. In Step (E9), we find $|F(A \cup \{v_7\}; G_r)|$



(a) An example of graph G_r.

(b) Graph G_r^A.

Figure 1: An illustration of the flow of the proposed algorithm for evaluating $\{|F(A \cup \{v\}; G_r)|;\, v \in V \setminus A\}$, where $r \in R_G$ and $A = \{v_1\}$.

$= 4$. In Step (E10), we set $U = \{v_4, v_5, v_6, v_7\}$. In Step (E11), we return to Step (E5). In Step (E5), we check $V_r^A \setminus U = \emptyset$. Then, the process of the algorithm ends.

4.3 Computational Complexity of Proposed Method

In the same way as in Section 3.3, we evaluate the computational complexity of the proposed method as the expected number of examined nodes for estimating all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A in Step (G3) of the greedy algorithm.

Let G_r be a graph generated from the occupation probability distribution $q(r)$ of the corresponding bond percolation model. We consider evaluating the expected number $\overline{Z(A, G_r)}$ of examined nodes for computing $\{|F(A \cup \{v\}; G_r)|;\, v \in V \setminus A\}$ by the proposed method (see, Section 4.2). First, the number of examined nodes for finding $F(A; G_r)$ is given by $|F(A; G_r)|$. Let

$$V_r^A = \bigcup_{u \in U_r^A} SCC(u; G_r^A)$$

be the SCC decomposition of the induced graph G_r^A of G_r to $V_r^A = V \setminus F(A; G_r)$, where U_r^A stands for the set of all the representative nodes for SCCs. For any $u \in U_r^A$, the number of examined nodes for finding $F(u; G_r^A)$ is $|F(u; G_r^A)|$. Suppose now that $F(u; G_r^A)$ is found. Then, the number of examined nodes for finding $C(u; G_r^A)$ $(= SCC(u; G_r^A))$ is $|SCC(u; G_r^A)|$, since $C(u; G_r^A) = B(u; G_r^A) \cap F(u; G_r^A)$ is calculated on the induced graph of graph G_r^A to $F(u; G_r^A)$. Therefore, the number $Z(A, G_r)$ of examined nodes for computing $\{|F(A \cup \{v\}; G_r)|;\, v \in V \setminus A\}$ by the proposed method is as follows:

$$Z(A, G_r) = |F(A; G_r)| + \sum_{u \in U_r^A} \left(|F(u; G_r^A)| + |SCC(u; G_r^A)| \right).$$

By the definition of graph G_r^A, we have

$$\sum_{u \in U_r^A} |SCC(u; G_r^A)| = N - |F(A; G_r)|,$$

where $N = |V|$. Thus, we have

$$Z(A, G_r) = N + \sum_{u \in U_r^A} |F(u; G_r^A)|. \tag{4}$$

Since $|F(u; G_r^A)| = |F(A \cup \{u\}; G_r)| - |F(A; G_r)|$, we can estimate the expected value of $|F(u; G_r^A)|$ as $\sigma(A \cup \{u\}) - \sigma(A)$. Hence, by Equation (4), we can estimate the expected number $\overline{Z(A, G_r)}$ of examined nodes for computing $\{|F(A \cup \{v\}; G_r)|;\, v \in V \setminus A\}$ as

$$\overline{Z(A, G_r)} = N + \left\langle \sum_{u \in U_r^A} (\sigma(A \cup \{u\}) - \sigma(A)) \right\rangle_r,$$

where $\langle f(r) \rangle_r$ stands for the operation that averages $f(r)$ with respect to r under $q(r)$, that is,

$$\langle f(r) \rangle_r = \sum_{r \in R(G)} f(r)\, q(r).$$

From the above results, we can estimate that the expected number $\mathcal{C}_1$ of examined nodes for the proposed method is

$$\mathcal{C}_1 = M \left\{ N + \left\langle \sum_{u \in U_r^A} (\sigma(A \cup \{u\}) - \sigma(A)) \right\rangle_r \right\}. \tag{5}$$

4.4 Computational Complexity Comparison

We compare the proposed method with the conventional method in terms of computational complexity. Both methods need M to be specified as a parameter, and we use the same value for both. We note that more coin-flips are used in the conventional method. In fact, if we think of a single run, i.e., any one of the M runs, the expected number of coin-flips for the conventional method is $O(|V|\sigma(v))$ for both the IC and LT models, whereas that for the proposed method is $O(|E|)$ for the IC model and $O(|V|)$ for the LT model. Note that in case of LT model for the proposed method, the coin-flip is realized by roulette for each node, i.e., picking at most one incoming link. However, if we focus on a single node v for initial activation from which to propagate the information, the number of coin-flips are $O(\sigma(v))$ for both the conventional and the proposed methods and for both the IC and the LT models because only the activated nodes (the expected number is $\sigma(v)$) are on the paths that lead to reachable nodes from v in the proposed

method. Thus by using the same value of M, both would estimate $\sigma(v)$ with the same accuracy in principle (see Appendix A). The biggest difference is that in the conventional method, when A is not empty, many of the coin-flips are redundant; that is, the diffusion process from A is repeatedly performed, whereas in the proposed method, no such repetition is made. This contributes to the stability of the proposed method. Below we begin by explaining the reason why we investigate the examined nodes to compare the proposed and the conventional methods.

First, we consider the case of IC model. Both the proposed and the conventional methods flip a coin with a bias $p_{u,v}$ on a link (u, v) to decide whether to propagate the information through the link (u, v) or not. Here, if we assume that all the coins are flipped in advance for the conventional method and ignore the computational complexity for flipping a coin and deciding whether or not to propagate the information, then for both the proposed and the conventional methods, the major computation is to trace forward or backward the links the information propagates and identify the nodes to visit. Therefore, we evaluate the computational complexities of the both methods for the IC model in terms of the expected number of examined nodes.

Next, we consider the case of LT model. For the proposed method, we ignore the computational complexity for the process of choosing at most one incoming link of each node in the original graph. For the conventional method, we ignore the computational complexity for the process of choosing the threshold θ_v of each node v in the original graph. Note that the proposed method performs the process M times, whereas the conventional method performs the process MN times. Moreover, for the conventional method, we further ignore the computational complexity for adding the weights from the neighboring active nodes to a node and deciding whether the node becomes active or not. Then, the major computation for the conventional method is to trace forward the links the information propagates and identify the nodes to visit. Therefore, we also evaluate the computational complexities of the both methods for the LT model in terms of the expected number of examined nodes.

Now, we compare the proposed and the conventional methods in terms of the expected number of examined nodes. We use the results in Sections 3.3 and 4.3. By Equation (2), the expected number $\mathcal{C}_0$ of examined nodes for the conventional method can be estimated as

$$\mathcal{C}_0 = M \left\{ N - |A| + \sum_{u \in V \setminus A} (\sigma(A \cup \{u\}) - 1) \right\}, \tag{6}$$

since $\sum_{V \setminus A} 1 = N - |A|$. In Equation (6), we can expect that $|A| \ll N$ $(= |V|)$, and $\sigma(A \cup \{u\}) - 1$ is summed up for almost all $u \in V$, since $k \ll N$. On the other hand, we can generally expect $|U_r^A| \ll N$ in Equation (5).

Also, we have $\sigma(A) > 1$ in the greedy algorithm if $A \neq \emptyset$. Moreover, for any $u \in V \setminus A$, $\sigma(A \cup \{u\}) - \sigma(A)$ decreases as $|A|$ increases, since $\sigma(A)$ is a submodular function. Hence, we can generally expect that in Step (G3) of the greedy algorithm, the proposed method has much smaller expected number of examined nodes than the conventional method.

From the above results, we can expect that compared with the conventional method, the proposed method will achieve a large reduction in computational cost.

5 Experimental Evaluation

Using large-scale real networks, we experimentally evaluated the performance of the proposed method.

5.1 Network Datasets

In the evaluation experiments, we should desirably use large-scale networks that exhibit many of the key features of real social networks. Here, we show the experimental results for two different datasets of such real networks.

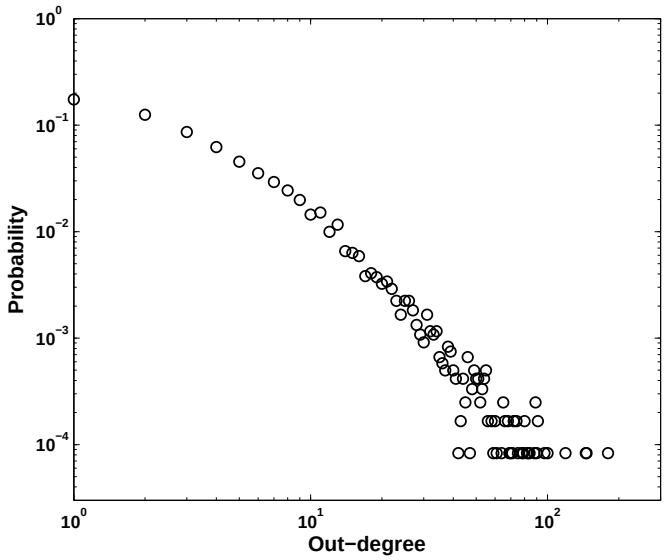


Figure 2: The out-degree distribution for the blog dataset.

First, we employed a trackback network of blogs, since a piece of information can propagate from one blog author to another blog author through a trackback, where a trackback is a kind of hyperlink with a *linkback* (i.e., link notification) function. We exploited the blog "Theme salon of blogs"

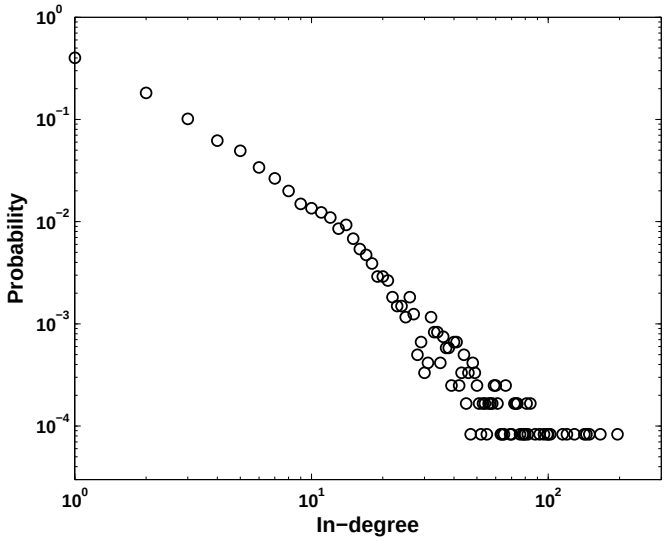


Figure 3: The in-degree distribution for the blog dataset.

in the site "goo" (`http://blog.goo.ne.jp/usertheme/`), where blog authors could recruit trackbacks of other blog authors by registering interesting themes. We collected a large-scale connected trackback network in May, 2005 by the following breadth first search process:

1. We started the process from the blog of the theme "JR Fukuchiyama Line Derailment Collision" in the site "goo", analyzed its HTML file, and extracted the list of the URLs of the source blogs of the trackbacks to this blog.

2. For each list obtained, we collected the blogs of the URLs in the list.

3. For each blog collected, we analyzed its HTML file, and constructed the list of the URLs of the source blogs of the trackbacks to the blog.

4. We repeated from Step 2 until depth ten from the original blog.

We call this network data the blog dataset. This network was a directed graph of $12,047$ nodes and $53,315$ links, and is expected to have a feature of real world social network in light of the way it is generated. To confirm this, the out-degree and in-degree distributions are plotted in Figures 2 and 3, from which it is understood that these are "heavy-tailed" distributions that most large real networks exhibit. Here, the out-degree and in-degree distributions are the distributions of the number of outgoing and incoming links for every node, respectively. Thus, we believe that the blog dataset is a typical example of a large real social network represented by a directed

graph, and can be used as the network data to evaluate the performance of the proposed method.

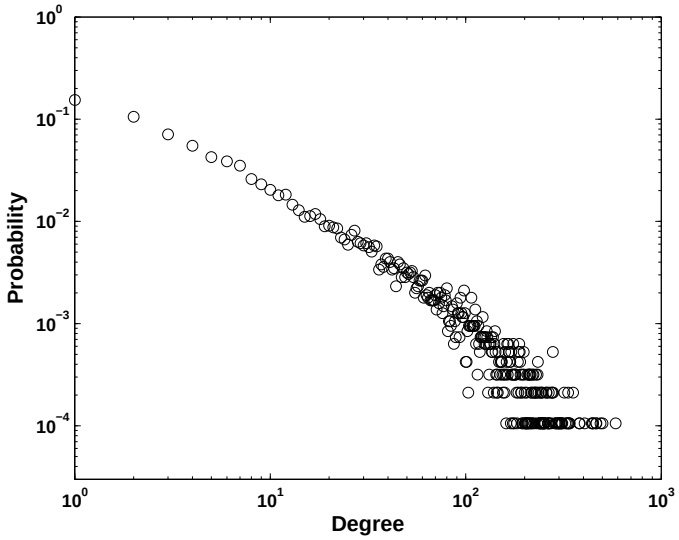


Figure 4: The degree distribution for the Wikipedia dataset.

Next, we employed a network of people that was derived from the "list of people" within Japanese Wikipedia. Specifically, we extracted the maximal connected component of the undirected graph obtained by linking two people in the "list of people" if they co-occur in six or more Wikipedia pages, and constructed a directed graph by regarding those undirected links as bidirectional ones. We call this network data the Wikipedia dataset. The total numbers of nodes and directed links were $9,481$ and $245,044$, respectively. Compared with the blog network, the way this network is generated is rather synthetically. Figure 4 shows the degree distribution of the undirected graph. We also observe that the degree distribution is a "heavy-tailed" distribution.

For social networks represented as undirected graphs, Newman and Park (2003) observed that they generally have the following two statistical properties that non-social networks do not have. First, they show positive correlations between the degrees of adjacent nodes. Second, they have much higher values of the *clustering coefficient* than the corresponding *configuration models* (i.e., random network models). Here, the clustering coefficient C for an undirected graph is defined by

$$C = \frac{3 \times \text{number of triangles on the graph}}{\text{number of connected triples of nodes}},$$

where a "triangle" means a set of three nodes each of which is connected to each other, and a "connected triple" means a node connected directly to

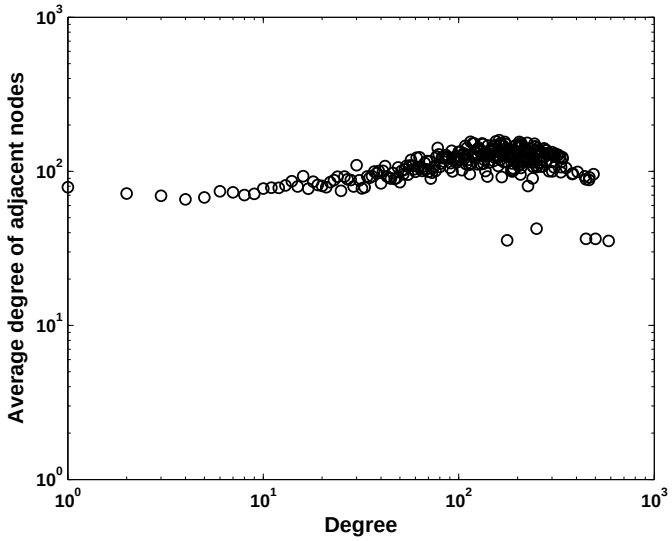


Figure 5: The degree correlation for the Wikipedia dataset.

unordered other pair nodes. Note that in terms of sociology, C measures the probability that two of your friends will also be friends each other. Given a degree distribution $\{\lambda_d\}$, the corresponding configuration model of a random network of N nodes is defined as the ensemble of all possible undirected graphs of N nodes that possess the degree distribution $\{\lambda_d\}$, where λ_d is the fraction of nodes in the network having degree d. It is known [18] that the value of C for the configuration model is exactly calculated by

$$C \;=\; \frac{1}{Nz_1}\left(\frac{z_2}{z_1}\right)^2,$$

where

$$z_1 \;=\; \sum_d \; d\lambda_d$$

is the average number of neighbors of a node and

$$z_2 \;=\; \sum_d \; d^2\lambda_d \;-\; \sum_d \; d\lambda_d$$

is the average number of second neighbors. For the undirected graph of the Wikipedia dataset, the value of C of the corresponding configuration model was 0.046, while the actual measured value of C was 0.39. Namely, the undirected graph of the Wikipedia dataset had a much higher value of the clustering coefficient than the corresponding configuration model. Moreover, we can see from Figure 5 that the Wikipedia dataset had weakly positive degree correlation. Therefore, we believe that the Wikipedia dataset is also

a typical example of a large real social network represented by an undirected graph, and can be used as the network data to evaluate the performance of the proposed method.

5.2 Experimental Settings

The proposed and the conventional methods are equipped with parameter M. We refer to the conventional method with $M = 1,000$ for the IC model as the *IC1000*. In the same way, we define the *LT1000* and *LT10000* for the conventional method with the LT model. We also refer to the proposed method using $M = 1,000$ and $M = 10,000$ for the IC model as the *ICBP1000* and *ICBP10000*, respectively. In the same way, we define the *LTBP1000* and *LTBP10000* for the proposed method with the LT model. As described in Section 4.4, we compare these methods for the same value of M.

The IC and LT models have parameters to be specified in advance. In the IC model, we assigned a uniform probability p to the propagation probability $p_{u,v}$ for any directed link (u, v) of the network, that is, $p_{u,v} = p$. In the LT model, we uniformly set weights as follows: For any node v of the network, the weight $w_{u,v}$ from a parent node $u \in \Gamma(v)$ is given by $w_{u,v} = 1/|\Gamma(v)|$.

We implemented all our programs of both the conventional and proposed methods for the IC and LT models in C language. Of course, the basic structure of these programs is the same, except that the routines of active node calculation used in the conventional method are replaced with those of bond percolation and SCC decomposition used in the proposed method.

5.3 Experimental Results

We compared the proposed method with the conventional method in terms of both the performance of the approximate solution A_k and the processing time for solving the influence maximization problem of size k. The performance of A_k is measured by the influence degree $\sigma(A_k)$. We estimated $\sigma(A_k)$ by using $300,000$ simulations according to the work of Kempe et al. (2003). All our experimentation was undertaken on a single Dell PC with an Intel 3.4GHz Xeon processor, with 2GB of memory, running under Linux.

In order to keep computational time at a reasonable level for the conventional method, we mainly compared these two methods using $M = 1,000$. Note that if a large enough M is taken, these two methods should produce the same solution. We conjecture that $M = 1,000$ is not large enough, that is, these two methods with $M = 1,000$ cannot necessarily obtain good approximate values for the marginal influence degrees $\{\sigma(A \cup \{v\});\ v \in V \setminus A\}$ of A, (see Appendices A and B). Thus, we iterated the same experiment five times independently. Tables 1 and 2 show the experimental results for the IC model with $p = 10\%$ and the LT model for the blog dataset, respectively, where the values are rounded to three significant figures. Note that

Table 1: Performance of approximate solutions for the influence maximization problem under the IC model with $p = 10\%$ for the blog dataset. Upper: IC1000 (the conventional method). Lower: ICBP1000 (the proposed method).

k	$\sigma(A_k)$ (IC1000)				
1	1.74×10^2	1.74×10^2	1.74×10^2	1.74×10^2	1.74×10^2
10	6.93×10^2	6.98×10^2	6.93×10^2	6.91×10^2	6.95×10^2
20	8.58×10^2	8.61×10^2	8.57×10^2	8.58×10^2	8.60×10^2
30	9.59×10^2	9.69×10^2	9.68×10^2	9.66×10^2	9.78×10^2

k	$\sigma(A_k)$ (ICBP1000)				
1	1.74×10^2	1.74×10^2	1.74×10^2	1.74×10^2	1.74×10^2
10	7.02×10^2	7.01×10^2	7.00×10^2	7.01×10^2	7.02×10^2
20	8.74×10^2	8.75×10^2	8.73×10^2	8.74×10^2	8.73×10^2
30	9.91×10^2	9.92×10^2	9.90×10^2	9.92×10^2	9.92×10^2

in these tables and later ones, too, the values are reestimated with $300,000$ simulations once A_k has been obtained by each method with a specified M. Since the true solution $\sigma(A_k^*)$ is by definition the maximum among all $\sigma(A_k)$, if $\sigma(A_k)$ is estimated accurately, it makes sense to argue that the larger the value is, the closer it is to the true solution and thus it is of better quality. We first observe that the results for the proposed method were relatively stable over the iterations, while the results for the conventional method somewhat fluctuated for large k in particular. Here, we note that the proposed method using $M = 10,000$ was stable and always produced the same solution for $k = 30$ over the iterations (not shown in the tables). We also observe that for $k = 30$, the solutions by the ICBP1000 and LTBP1000 outperforms those by the IC1000 and LT1000, respectively.

Table 3 shows the processing time to obtain A_k by the IC1000, ICBP1000, LT1000 and LTBP1000 for the blog dataset, where the values are rounded to three significant figures. We observe from Table 3 that the ICBP1000 and LTBP1000 are much more efficient than the IC1000 and LT1000, respectively. For example, to obtain the approximate solution A_{30} for $k = 30$, both the IC1000 and LT1000 needed about 2.5 days, while the ICBP1000 and LTBP1000 needed about 2.5 and 1.5 minutes, respectively. Namely, for $k = 30$, the ICBP1000 was 1.8×10^3 times faster than the IC1000, and the LTBP1000 was 4.6×10^3 times faster than the LT1000. We also examined the LT10000 and LTBP10000 on the blog dataset. In order to obtain approximate solution A_{30}, the LT10000 needed about 27 days, while the LTBP10000 needed only about 14 minutes.

Table 2: Performance of approximate solutions for the influence maximization problem under the LT model for the blog dataset. Upper: LT1000 (the conventional method). Lower: LTBP1000 (the proposed method).

k	$\sigma(A_k)$ (LT1000)				
1	2.86×10^2	2.86×10^2	2.86×10^2	2.86×10^2	2.86×10^2
10	1.59×10^3	1.61×10^3	1.61×10^3	1.59×10^3	1.58×10^3
20	2.41×10^3	2.40×10^3	2.42×10^3	2.42×10^3	2.38×10^3
30	3.02×10^3	3.05×10^3	3.01×10^3	3.01×10^3	3.00×10^3

k	$\sigma(A_k)$ (LTBP1000)				
1	2.86×10^2	2.86×10^2	2.86×10^2	2.86×10^2	2.86×10^2
10	1.60×10^3	1.61×10^3	1.61×10^3	1.59×10^3	1.60×10^3
20	2.44×10^3	2.44×10^3	2.44×10^3	2.44×10^3	2.44×10^3
30	3.07×10^3	3.07×10^3	3.06×10^3	3.06×10^3	3.06×10^3

Table 3: Processing time (sec.) for the blog dataset.

k	IC1000	ICBP1000	LT1000	LTBP1000
1	3.70×10^2	7.07	6.57×10^2	3.19
10	4.69×10^4	5.68×10^1	4.24×10^4	2.96×10^1
20	1.24×10^5	1.09×10^2	1.25×10^5	5.64×10^1
30	2.13×10^5	1.60×10^2	2.32×10^5	8.20×10^1

Tables 4, 5 and 6 show the experimental results for the Wikipedia dataset. We see that the results were qualitatively very similar to the ones for the blog dataset. First, the solutions by the ICBP1000 and LTBP1000 outperformed those by the IC1000 and LT1000, respectively. We also note that the proposed method using $M = 10,000$ was stable and always produced the same solution for $k = 30$ over the iterations (not shown in the tables). Next, the ICBP1000 and LTBP1000 were much more efficient than the IC1000 and LT1000, respectively. For example, for obtaining the approximate solution A_{30} for $k = 30$, the ICBP1000 was 1.9×10^3 times faster than the IC1000, and the LTBP1000 was 8.3×10^3 times faster than the LT1000. We also conducted experiments on some other large-scale real networks including a blogroll network of blogs, and confirmed the effectiveness of the proposed method.

Table 4: Performance of approximate solutions for the influence maximization problem under the IC model with $p = 1\%$ for the Wikipedia dataset. Upper: IC1000 (the conventional method). Lower: ICBP1000 (the proposed method).

k	$\sigma(A_k)$ (IC1000)				
1	1.39×10^2	1.39×10^2	1.36×10^2	1.36×10^2	1.36×10^2
10	3.91×10^2	3.97×10^2	3.98×10^2	4.02×10^2	4.01×10^2
20	4.56×10^2	4.64×10^2	4.62×10^2	4.64×10^2	4.66×10^2
30	4.97×10^2	5.02×10^2	4.95×10^2	5.00×10^2	4.98×10^2

k	$\sigma(A_k)$ (ICBP1000)				
1	1.39×10^2	1.39×10^2	1.39×10^2	1.36×10^2	1.36×10^2
10	4.05×10^2	4.06×10^2	4.07×10^2	4.06×10^2	4.07×10^2
20	4.75×10^2	4.76×10^2	4.76×10^2	4.75×10^2	4.77×10^2
30	5.16×10^2	5.17×10^2	5.17×10^2	5.16×10^2	5.17×10^2

5.4 Discussion

These experimental results show that the proposed method is much more efficient than the conventional method.

First, we investigate the reason why the proposed method outperforms the conventional method in the case of $M = 1,000$ for our network datasets. If we take a sufficiently large M (e.g., $M = 100,000$), the proposed and the conventional methods should produce the same solution. As shown in the experiments, the estimation accuracy of influence degree function σ with $M = 1,000$ is not so high for the both methods. Now, consider estimating all the marginal influence degrees $\{\sigma(A_k \cup \{v\});\ v \in V \setminus A_k\}$ of solution A_k, and choosing the node v_{k+1} that maximizes $\sigma(A_k \cup \{v\})$, $(v \in V \setminus A_k)$. It should be reemphasized that the influence set of A_k is equally evaluated for all $v \in V \setminus A_k$ for the proposed method. In fact, when $\sigma(A_k \cup \{v\})$ is estimated using Equation (3), each $|F(A_k \cup \{v\}; G_{r_m})|$ is basically computed by

$$|F(A_k \cup \{v\}; G_{r_m})| = \left|F(v; G_{r_m}^{A_k})\right| + |F(A_k; G_{r_m})|\,.$$

Thus, for the proposed method, a node that is relatively optimal for A_k can be selected as v_{k+1}. On the other hand, for the conventional method, the influence set of A_k is not equally evaluated for all $v \in V \setminus A_k$ since $\sigma(A_k \cup \{v\})$ is independently estimated for every v each by a distinct simulation. We also note that the number of final active nodes for a given target set greatly varied for every simulation in the IC and LT models (see, Appendix B). Thus, unlike the proposed method, the selection of v_{k+1} in the conventional method

Table 5: Performance of approximate solutions for the influence maximization problem under the LT model for the Wikipedia dataset. Upper: LT1000 (the conventional method). Lower: LTBP1000 (the proposed method).

k	$\sigma(A_k)$ (LT1000)				
1	3.41×10^2	3.41×10^2	3.41×10^2	3.41×10^2	3.41×10^2
10	1.72×10^3	1.72×10^3	1.67×10^3	1.66×10^3	1.72×10^3
20	2.55×10^3	2.55×10^3	2.45×10^3	2.53×10^3	2.55×10^3
30	3.12×10^3	3.03×10^3	2.99×10^3	3.01×10^3	3.11×10^3

k	$\sigma(A_k)$ (LTBP1000)				
1	3.41×10^2	3.41×10^2	3.41×10^2	3.41×10^2	3.41×10^2
10	1.72×10^3	1.72×10^3	1.72×10^3	1.72×10^3	1.71×10^3
20	2.58×10^3	2.58×10^3	2.59×10^3	2.59×10^3	2.59×10^3
30	3.18×10^3	3.18×10^3	3.18×10^3	3.18×10^3	3.18×10^3

Table 6: Processing time (sec.) for the Wikipedia dataset.

k	IC1000	ICBP1000	LT1000	LTBP1000
1	6.63×10^2	1.91×10^1	5.41×10^2	5.17
10	1.94×10^5	1.74×10^2	9.60×10^4	4.64×10^1
20	4.82×10^5	3.42×10^2	3.03×10^5	8.57×10^1
30	8.03×10^5	5.10×10^2	5.69×10^5	1.21×10^2

using $M = 1,000$ by necessity completely depends on how the influence set of A_k is evaluated by chance for each $v \in V \setminus A_k$. Therefore, we believe that the proposed method outperforms the conventional method in the case of $M = 1,000$ for our network datasets.

Here, to explain the point of the reason described above more clearly, we consider the following method as an extended version of the conventional method:

1. **for** $m = 1$ to M **do**
2. Find the set $D(A_k)$ of active nodes at the end of the random process of the IC or the LT models for initial active set A_k by simulation.
3. **for** each $v \in V \setminus A_k$ **do**
4. Find the set $D(v)$ of active nodes at the end of the random process of the IC or the LT models for initial active set $\{v\}$ by simulation.

5. Set $x_{v,m} \leftarrow |D(A_k) \cup D(v)|$.

6. **end for**

7. **end for**

8. **for** each $v \in V \setminus A_k$ **do**

9. Set $\sigma(A_k \cup \{v\}) \leftarrow (1/M) \sum_{m=1}^{M} x_{v,m}$

10. **end for**

The extended method should improve the conventional method because the influence set of A_k is now equally evaluated for all $v \in V \setminus A_k$, and should be comparable to the proposed method in quality of solution. However, it cannot be as efficient as the proposed method since it does not incorporate the SCC-finding technique.

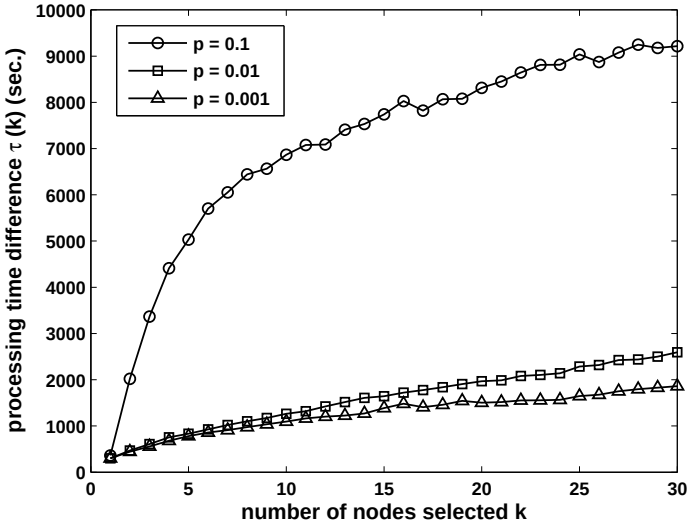


Figure 6: Processing time difference $\tau(k)$ between the proposed and conventional methods for the blog dataset in the case of the IC model.

Next, we discuss the sources of the difference between the proposed and conventional methods in processing time. Note that we use the same value of parameter M for both methods. Let $\tau_1(k)$ and $\tau_0(k)$ respectively denote the processing times of the proposed and the conventional methods for obtaining solution A_{k+1} when solution A_k is given. We define the processing time difference $\tau(k)$ by $\tau_0(k) - \tau_1(k)$ for k, the number of nodes selected. We believe the essential sources of speed-up in the proposed method is that we compute $\{|F(A_k \cup \{v\}; G_r)|;\, v \in V \setminus A_k\}$ on graph G_r as follows:

- By first identifying $F(A_k; G_r)$, we reduce the graph in question from G_r to the induced graph $G_r^{A_k}$ of G_r to $V \setminus F(A_k; G_r)$
- By decomposing $G_r^{A_k}$ into the SCCs, we compute $|F(A_k \cup \{v\}; G_r)|$ for many nodes v at once.

Namely, we believe that the larger the size of $F(A_k; G_r)$ is, the larger the value of $\tau(k)$ is. Moreover, we believe that the larger the sizes of the SCCs of graph $G_r^{A_k}$ are, the larger the value of $\tau(k)$ is. Here, we demonstrate these characteristics for the IC model. Note that the size of $F(A_k; G_r)$ monotonically increases with the value of k. Thus, we can expect that the value of $\tau(k)$ also monotonically increases with the value of k. Note also that graph G_r becomes denser when the value of the propagation probability p is larger, and the sizes of the SCCs of G_r also become larger. Thus, we can also expect that the value of $\tau(k)$ monotonically increases with the value of p. Figure 6 shows $\tau(k)$ for $p = 0.1\%$, 1% and 10% as a function of k for the blog dataset, where circles, squares and diamonds indicate $\tau(k)$ for $p = 0.1\%$, 1% and 10%, respectively. Here, we used $M = 1,000$ for both the proposed and the conventional methods. The results support our conjectures.

6 Related Work

6.1 Calculation of Influence Degrees

First, we describe work related to the calculation of influence degrees in the IC model. Let us recall that the SIR model for the spread of a disease on a network is equivalent to a bond percolation model on the same network, and the size of a disease outbreak from a node corresponds to the size of the cluster that can be reached from the node by traversing only the "occupied" links. There are a series of work that uses this correspondence to develop a method for theoretically calculating the probability distribution of the size of a disease outbreak that starts with a randomly chosen node in the configuration model (i.e., a random network model) with a given degree distribution (Callaway et al., 2000; Newman, 2002; Newman, 2003), and to derive a condition for the disease outbreak from a randomly chosen node to give an *epidemic outbreak* that affects a non-zero fraction on the network in the limit of very large network. Mathematically more rigorous treatments of similar results can be found in the work of Molloy and Reed (1998) and Chung and Lu (2002).

Next, we describe work related to the calculation of influence degrees in the LT model. Watts (2002) investigated the LT model on a network to explain large but rare cascade phenomena triggered by small initial shocks. Using the concept of *site percolation*, he theoretically derived a condition for the cascade from a randomly chosen seed node to give a *global cascade* that affects a non-zero fraction on the network in the limit of infinitely large

network for the configuration model (i.e., a random network model) with a given degree distribution.

The above mentioned studies focused on global properties averaged over a random network in the limit of very large size, while our primary interest is to practically answer which nodes are most influential for information diffusion on a given real-world network of a finite size. We also note that those studies dealt with undirected graphs, while our work investigates information diffusion on networks represented by directed graphs. Moreover, the theories developed in those studies assumed that the loop structure on a network of interest can be essentially ignored in the limit of large network size. However, this property is not true of many large-scale social networks, and it is an open question whether or not those theories are effective for such networks (Newman, 2003). In fact, the clustering coefficient C quantifies the loop structure in a network, and it was indeed observed that many social networks have much higher values of C than the corresponding configuration models (i.e., random network models) (Newman and Park, 2003).

6.2 Solving the Influence Maximization Problem

The influence degree function σ is submodular (see, Kempe et al., 2003). For solving a combinatorial optimization problem of a submodular function f on V by the greedy algorithm, Leskovec et al. (2007) have recently presented a lazy evaluation method that leads to far fewer (expensive) evaluations of the marginal increments $f(A \cup \{v\}) - f(A)$ $(v \in V \setminus A)$ in the greedy algorithm for $A \neq \emptyset$, and achieved an improvement in speed. Note here that their method requires evaluating $f(v)$ for all $v \in V$ at least. Thus, we can apply their method to the influence maximization problem for the IC or LT models, where the influence degree function σ is evaluated through the simulations of the corresponding random process. It is clear that this method is more efficient than the conventional method. However, the proposed method for $k = 30$ was faster than the conventional method for $k = 1$ as shown in Tables 3 and 6. Therefore, it is evident that the proposed method can be faster than the method by Leskovec et al. (2007) for the influence maximization problem for the IC or LT models. To quantify the difference we implemented the Lazy evaluation method. The processing time for $k = 30$ in case of the blog dataset was 2.12×10^3 and 8.28×10^2 seconds for the IC and the LT models, respectively, and the corresponding processing time in case of Wikipedia dataset was 1.46×10^4 and 2.65×10^3 seconds for the IC and the LT models, respectively. Here, $M = 1,000$ are used as the number of simulations (see, Section 3.2), and the values are rounded to three significant figures. From these results, we can see that the proposed method was more than ten times faster than the method by Leskovec et al. (2007) for $k = 30$ in the blog and Wikipedia datasets (see, Tables 3 and 6).

Beyond the IC and LT models, Kempe et al. (2003) proposed the *trig-*

gering model as an yet another diffusion model on a network. It is proved that the triggering model can be identified with a bond percolation model (see, Kempe et al., 2003). The proposed method can be applied to this model because it can be applied to any diffusion model that can be identified with a bond percolation model. The future work includes presenting a large number of realistic examples of such diffusion models.

In this paper, we have considered the *progressive* case in which nodes cannot switch from being active to being inactive. However, there are many information diffusion phenomena that non-progressive diffusion models are required. Examples include the spread of posts for a topic in blogspace (Gruhl et al, 2004). Kempe et al. (2003) proved that *non-progressive* case can be reduced to the progressive case. More specifically, it is proved that the influence maximization problem for a non-progressive diffusion model on graph G in time-limit T is equivalent to the ordinary influence maximization problem on the *layered graph* G_T for the progressive diffusion model, where G_T is the directed acyclic graph (DAG) constructed by time-forwardly connecting $(T+1)$ copies of G (see, Kempe et al. 2003). Therefore, building effective methods for fundamental progressive models such as the IC and LT models is indeed important and crucial for the non-progressive case.

From a realistic point of view, the IC and LT models are by no means a complete model, but are at best a simplified and partial representation of a complex reality (see, Kempe et al, 2003; Gruhl et al., 2004; Leskovec et al., 2006). However, in the field of sociology, Watts and Dodds (2007) recently examined the "influentials hypothesis" in the contexts of the LT model and the SIR model (i.e., an extended model of the IC model), that is, they investigated by computer simulations whether large cascades of influence are actually driven by influentials or not. On the other hand, Even-Dar and Shapira (2007) mathematically studied the influence maximization problem in the context of another fundamental model called the voter model. We also believe that it is important to investigate information diffusion phenomena for the IC and LT models (i.e., fundamental diffusion models) to deepen our understanding of these models. The future work includes proposing effective methods for solving the influence maximization problem in the contexts of various realistic diffusion models.

6.3 Applications

As is easily understood, the conventional method is not practical unless we rely on high-performance computers and sophisticated techniques such as parallel computing (see, Tables 3 and 6) to solve the kind of problems such as influence maximization problem as addressed in this paper. In contrast, the proposed method enables us to obtain a practical solution to this kind of problems on a single standard PC in a reasonable processing time. Thus, we can apply the proposed method to a variety of real problems.

The work of Watts and Dodds (2007) briefly described above needs a method to efficiently estimate $\sigma(A)$ and the proposed method can readily be applicable.

As mentioned in the introduction, the influence maximization problem finds many realistic applications. The most straightforward application would be viral marketing. When we wish to promote a new product (e.g., an email service or a search engine), and are given a relevant social network, we can easily find a limited number of key (influential) persons first to adopt the new product by the proposed method, and enjoy the diffusion effect for the IC or LT models (i.e., fundamental diffusion models) through the social network. We admit that the diffusion models we discussed are oversimplified but still it is useful to obtain approximate solutions as a first step toward an effective marketing without using classical advertising channels.

The proposed method has an application of different flavor which is the visualization of information flow. Understanding the flow of information through a complex network is important in terms of sociology and marketing. We devised a new node embedding method for visualizing the information diffusion process from the target nodes selected to be a solution of the influence maximization problem (Saito et al., 2008). This visualization method is characterized by 1) utilization of the target nodes as a set of pivot objects for visualization, 2) application of a probabilistic algorithm for embedding all the nodes in the network into an Euclidean space to conserve the posterior information diffusion probability, and 3) varying appearance of the embedded nodes on the basis of two label assignment strategies, one with emphasis on influence of initially activated nodes, and the other on degree of information reachability.

7 Conclusion

We have considered the influence maximization problem for the IC and LT models on a large-scale social network represented as a directed graph $G = (V, E)$. Due to the computational complexity, the greedy search algorithm is the only practical approach, but still the conventional method needed a high amount of computation. We have proposed a method of efficiently finding a good approximate solution to the problem under the greedy algorithm. In particular, in order to improve the computational efficiency, we have estimated all the marginal influence degrees $\{\sigma(A \cup \{v\});\ v \in V \setminus A\}$ of a given target set A in the following way:

- We identify the IC and LT models with the corresponding bond percolation models.
- For any $v \in V \setminus A$, we estimate the influence degree $\sigma(A \cup \{v\})$ of $A \cup \{v\}$ as the empirical mean of the number $|F(A \cup \{v\}; G_r)|$ of the

nodes that are reachable from $A \cup \{v\}$ on a graph G_r generated from the corresponding occupation probability distribution $q(r)$ of the bond percolation.

In particular, we estimate $\{|F(A \cup \{v\}; G_r)|;\ v \in V \setminus A\}$ as follows:

- We find the set $F(A; G_r)$ that is reachable from A on graph G_r, and simultaneously compute $\{|F(A \cup \{v\}; G_r)|;\ v \in F(A; G_r)\}$.
- We find the induced graph G_r^A of G_r to $V \setminus F(A; G_r)$, and decompose G_r^A into its SCCs (Strongly Connected Components).
- For each SCC $SCC(u; G_r^A)$ of G_r^A, $(u \in V \setminus F(A; G_r))$, we simultaneously compute $\{|F(A \cup \{v\}; G_r)|;\ v \in SCC(u; G_r^A)\}$.

We have compared the proposed method with the conventional method in terms of computational complexity and quality of the solution, and have shown that the proposed method is expected to achieve a large amount of reduction in computational cost. Moreover, using large-scale networks including a real blog network, we have experimentally demonstrated the effectiveness of the proposed method. For example, we obtained the following results for the influence maximization problem of size $k = 30$ on the blog and Wikipedia datasets that are real networks with about $10,000$ nodes: In the case of the IC model, the proposed method was 1800 times faster than the conventional method, and in the case of the LT model, the proposed method was 4600 times faster than the conventional method.

Acknowledgement

This work was partly supported by JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147), and Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027.

Appendix

A Convergence Speed

As described in Section 4.4, by using the same value of M, both the proposed and the conventional methods would estimate $\sigma(v)$ with the same accuracy in principle. Here, we experimentally demonstrate this conjecture.

According to the work of Kempe et al. (2003), we set $M = 300,000$ as a sufficiently large value of M, that is, we assume that $\sigma(v)$ for any $v \in V$ is well approximated by $300,000$ simulations of the information diffusion model (i.e., the conventional method using $M = 300,000$). For any $v \in V$, let $\sigma_0(v; M)$ and $\sigma_1(v; M)$ denote the estimates of $\sigma(v)$ by the conventional and the proposed methods using parameter value M, respectively. For the blog and Wikipedia datasets, we investigated

$$\mathcal{E} = \frac{1}{N} \sum_{v \in V} |\sigma_0(v; 300, 0000) - \sigma_1(v; 300, 000)|,$$

$$\mathcal{E}_0(M) = \frac{1}{N} \sum_{v \in V} |\sigma_0(v; M) - \sigma_0(v; 300, 000)|,$$

$$\mathcal{E}_1(M) = \frac{1}{N} \sum_{v \in V} |\sigma_1(v; M) - \sigma_1(v; 300, 000)|.$$

We first consider the case of the IC model. Then, the value of $\mathcal{E}$ was 0.03 and 0.04 for the blog and Wikipedia datasets, respectively. Thus, we can assume that the values of $\sigma_0(v; 300, 000)$ and $\sigma_1(v; 300, 000)$ are almost the same for any $v \in V$.

Table 7: Convergence speed for the blog dataset.

M	$\mathcal{E}_0(M)$	$\mathcal{E}_1(M)$
100	1.16	1.12
1,000	0.36	0.36
10,000	0.11	0.12
100,000	0.03	0.03

Table 8: Convergence speed for the Wikipedia dataset.

M	$\mathcal{E}_0(M)$	$\mathcal{E}_1(M)$
100	1.28	1.23
1,000	0.42	0.42
10,000	0.13	0.14
100,000	0.03	0.03

Tables 7 and 8 show the values of $\mathcal{E}_0(M)$ and $\mathcal{E}_1(M)$ for the blog and Wikipedia datasets, respectively. These results imply that the proposed and the conventional methods estimate $\{\sigma(v);\, v \in V\}$ with almost the same

accuracy for the IC model. We also obtained similar results for the case of the LT model. For example, the value of $\mathcal{E}$ was 0.03 and 0.09 for the blog and Wikipedia datasets, respectively. For the blog dataset, the values of $\mathcal{E}_0(10,000)$ and $\mathcal{E}_1(10,000)$ were 0.13 and 0.12, respectively. Also, for the Wikipedia datasets, the values of $\mathcal{E}_0(10,000)$ and $\mathcal{E}_1(10,000)$ were 0.36 and 0.37, respectively. These results support our conjecture.

B Fluctuation in Simulations of Information Diffusion Models

For each $v \in V$, we examine fluctuation in the number $\varphi(v)$ of the final active nodes for a target initially activated node v through $1,000$ simulations in the IC and LT models. Let $\mu(v)$ and $s(v)$ denote the empirical mean and the standard deviation of $\varphi(v)$ for $1,000$ simulations, respectively. We define $\overline{\mu}$ and $\overline{s}$ by the empirical means of $\{\mu(v);\ v \in V\}$ and $\{s(v);\ v \in V\}$, respectively. For the blog dataset, $\overline{\mu}$ and $\overline{s}$ were as follows:

IC model ($p = 10\%$): $\overline{\mu} = 8.6$, $\overline{s} = 14.3$.

LT model: $\overline{\mu} = 6.8$, $\overline{s} = 14.9$.

For the Wikipedia dataset, $\overline{\mu}$ and $\overline{s}$ were as follows:

IC model ($p = 1\%$): $\overline{\mu} = 8.1$, $\overline{s} = 16.1$,

LT model: $\overline{\mu} = 12.6$, $\overline{s} = 42.4$,

Here, the values are rounded to the first decimal place. We can observe that compared with $\overline{\mu}$, $\overline{s}$ is very large. Therefore, we see that the number of final active nodes for a given target set can greatly vary for every simulation in the IC and LT models.

References

[1] Callaway, D. S., Newman, M. E. J., and Strogatz, S. H. 2000. Network robustness and fragility: Percolation on random graphs. Physical Review Letters, 85:5468–5471.

[2] Chung, F. and Lu, L. 2002. Connected components in a random graph with given expected degree sequences. Annals of Combinatorics, 6:125-145.

[3] Domingos, P. 2005. Mining social networks for viral marketing. IEEE Intelligent Systems, 20:80–82.

[4] Domingos, P. and Richardson, M. 2001. Mining the network value of customers. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, pp. 57–66.

[5] Even-Dar, E. and Shapira, A. 2007. A note on maximizing the spread of influence in social networks. Internet and Network Economics: WINE 2007, LNCS 4858, pp. 281–286.

[6] Goldenberg, J., Libai, B., and Muller, E. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters, 12:211–223.

[7] Grassberger, P. 1983. On the critical behavior of the general epidemic process and dynamical percolation. Mathematical Bioscience, 63:157–172.

[8] Gruhl, D., Guha, R., Liben-Nowell, D., and Tomkins, A. 2004. Information diffusion through blogspace. Proceedings of the 7th International World Wide Web Conference, New York, USA, pp. 107–117.

[9] Kempe, D., Kleinberg, J., and Tardos, E. 2003. Maximizing the spread of influence through a social network. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, pp. 137–146.

[10] Kempe, D., Kleinberg, J., and Tardos, E. 2005. Influential nodes in a diffusion model for social networks. Automata, Languages and Programming: ICALP 2005, LNCS 3580, pp. 1127–1138.

[11] Leskovec, J., Adamic, L. A., and Huberman, B. A. 2006. The dynamics of viral marketing. Proceedings of the 7th ACM Conference on Electronic Commerce, Ann Arbor, Michigan, USA, pp. 228–237.

[12] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. 2007. Cost-effective outbreak detection in networks. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, pp. 420–429.

[13] McCallum, A., Corrada-Emmanuel, A., and Wang, X. 2005. Topic and role discovery in social networks. Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburugh, Scotland, pp. 786–791.

[14] Molloy, M. and Reed, B. 1998. The size of the giant component of a random graph with a given degree sequence. Combinatorics, Probability and Computing, 7:295–305.

[15] Newman, M. E. J. and Park, J. 2003. Why social networks are different from other types of networks. Physical Review E, 68:036122.

[16] Newman, M. E. J. 2001. The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences of the United States of America, 98:404–409.

[17] Newman, M. E. J. 2002. Spread of epidemic disease on networks. Physical Review E, 66:016128.

[18] Newman, M. E. J. 2003. The structure and function of complex networks. SIAM Review, 45:167–256.

[19] Richardson, M. and Domingos, P. 2002. Mining knowledge-sharing sites for viral marketing. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, pp. 61–70.

[20] Saito, K., Kimura, M., and Motoda, H. 2008. Effective visualization of information diffusion process over complex networks. Machine Learning and Knowledge Discovery in Databases: ECML PKDD 2008, LNAI 5212, pp. 326–341.

[21] Watts, D. J. 2002. A simple model of global cascades on random networks. Proceedings of the National Academy of Sciences of the United States of America, 99:5766–5771.

[22] Watts, D. J. and Dodds, P. S. 2007. Influence, networks, and public opinion formation. Journal of Consumer Research, 34:441–458.